

**Dokumentation**  
der Kolloquiums zu Ehren von  
Prof. Dr. e.h. Konrad Zuse

10. Oktober 1979

Paul E. Ceruzzi	The early work of Konrad Zuse - a historian's view	2
Eike Jessen	Konrad Zuse - Konstrukteur der ersten programmgesteuerten Rechenanlage	11
Friedrich L. Bauer	Formalisierung und Programmierung bei Konrad Zuse	14
Hermann Flessner	Konrad Zuse - Sein Wirken allein und mit anderen	26
	Urkunde zur Ehrenpromotion	31
Konrad Zuse:	Die Emanzipation der Datenverarbeitung	32
Konrad Zuse	Emancipation of Data Processing (English Translation)	40

Fachbereich Informatik, Universität Hamburg

## **The Early Work of Konrad Zuse -- an Historian's View**

**Paul E. Ceruzzi**

4557 Forest Ave., Kansas City, MO 64110, USA

Presented at the awarding of the honorary Doctoral Degree  
to Herrn Prof. Dr-Ing. Konrad Zuse,  
October 10, 1979, Hamburg, West Germany.

### **The Early work of Konrad Zuse -- an Historian's View**

First of all I would like to say that I am honored to have been chosen to give the opening address at this colloquium, and I want to thank the **Institut für Informatik** of the University of Hamburg not only for their gracious invitation, but also for their decision to honor Herr Doktor Zuse in this way. By their awarding to Konrad Zuse the honorary doctoral degree, they are reaffirming the central fact that science is a cumulative and positive activity of mankind. By all standards of measurement, the progress of computer science in the past few decades has been astounding, but we must always remember that we can enjoy that progress only because we have been able to build on the work of pioneers like Dr. Zuse who came before us.

Precisely because science is a cumulative and positive activity, the historian must approach the subject in a very different way than would a working scientist. The scientist builds on the successes of those before him, but the study of history is primarily a study of individuals, of the totality of their work, and it is not possible to separate the successes from the failures in an individual's work and still remain fair to the historical evidence. Today we know that certain avenues of computer development were dead-ends, in that they are little-used, or have been superseded by something better, or abandoned as impractical. The historian, living and working in the present, is obviously aware of the course of events, even if he is not an expert in modern computer theory or programming. But in his assessment of the past record he must try to judge the past on its own terms, not on the basis of what would later become successful. The person who was making those discoveries obviously could not have known in which directions the future course of events would take. If he did know, there would not have been any discovery at all. In addition, what we accept as correct today may very well be considered absolutely **wrong** a few years hence.<sup>1</sup>

The tendency for historians to bias their assessment of the past by applying the standards of the present has often been discussed, and is sometimes referred to as the "Whig Interpretation" of history, referring to a generation of British historians in the last century.<sup>2</sup> I mention it at the beginning of this address because I feel that it is especially applicable to the study of the history of computing, which is itself such a new addition to the comparatively new field of history of science.

The Computer industry today is large by any standards. In the universities, the study of computer science is as active as that of many older, more traditional sciences. The very success of computers today increases the difficulty of maintaining a perspective on its history. Today, school children at the **gymnasium** level or younger can learn how a computer works and how to program it. But this does not mean that the discovery of those ideas was so simple! On the contrary, it was the very simplicity of those ideas which made them difficult to discover. Later in this address, I will show how I feel that one of Herr Zuse's greatest achievements was his discovery that the logical design

of a digital computer was based on the three fundamental operations of the symbolic logic -- Conjunction, Disjunction, and Negation. It is a simply idea that we can all understand today, yet it eluded many of Zuse contemporaries who had also sought a basis for computer design.

The historian, than has to try to recapture the **milieu** in which the computer was invented. He is of course aware of which ideas were later to be successful, and he may indicate this to his audience if he wants to. But this is not his main goal. At the same time, the historian does not have to have the same kind of mastery of the subject that the contemporary computer scientist has. He does not have to concern himself with how modern integrated circuits are produced, but he is concerned with the operation of vacuum tubes, mechanical linkages, and relays -- things that today's computer scientist probably never has to think of.

### **Konrad Zuse's Early Work**

One of the most interesting aspects of Herr Zuse's work is that it arose independently of the commercial calculating machine industry, the business machines industry, and the manufacturers of scientific instruments. In his autobiography, **Der Computer. Mein Lebenswerk**, Herr Zuse mentions how his first thoughts on how to mechanize calculation were "**etwas kindlich-primi-tiv**."<sup>3</sup> At that time, commercial firms such as Brunsviga and Ohdner were making decimal mechanical calculators which were compact, reliable, and well-designed. Bookkeeping machines which used the Hollerith punched card were also in wide use by 1950. And the scientific community had developed complex and powerful analog machines, including sophisticated versions of the familiar slide-rule, to facilitate their work.<sup>4</sup>

But Herr Zuse's work was to follow an entirely independent course. From the beginning he chose to build digital, not analog, machines, and he chose the binary rather than decimal scale to represent numbers internally. Both of these decisions represent a strong break with the prevailing ideas of the day, not only in Germany but elsewhere as well.<sup>5</sup>

In this sense we can say the the history of the computer began in the 1930's when pioneers like Dr. Zuse began building machines that were entirely different from what had come before. But if the history of the computer is to have any real meaning, it must be shown to be a part of the much larger field of the history of science and technology, especially mathematics. Therefore a proper history of the computer would begin with man's first attempts to count, record, and manipulate numbers. It would begin with a study of the ancient civilizations of Egypt, Mesopotamia, and India. In contrast to most other areas of the history of the mathematical sciences, we would find the Greeks mentioned only briefly. They had a knowledge of how to manipulate numbers, but the study of computation for its own sake was never pursued by them.

Now it is not possible in this brief address to give this kind of complete account, but I do want to mention some of the more immediate influences on the events of the 1930's. In particular, I would like to show how the ideas of Leibnitz (*correct is Leibniz!*), more than anyone else, have given us the foundation of modern computer science.

I have already mentioned how one of the most fascinating aspects of Herr Zuse's work is that from the beginning, he sought and developed a logical calculus which could describe both the arithmetic and control operations of his machines. In the course of his work he in fact rediscovered the **Aussagenkalkül** (symbolic logic) that Hilbert and Bernays had worked out earlier. Zuse called his calculus the "Conditional Combinatoric."

Such a unified approach to computer design was lacking in all of the immediate ancestors to Zuse, but it clearly resembles the ideas set out by Leibnitz in the late 17th Century. In his **Rechnen mit**

**Null und Eins**, for example, Leibnitz showed how ordinary arithmetic could be carried out in the binary system. In some of his other writings, he describes a universal language in which the truth or falsehood of a statement could be established by applying absolute mathematical rules.

Now to be sure, Leibnitz never connected this work on a logical language to binary arithmetic. Nor did he go beyond only a brief discussion of how to mechanize these processes (He did, of course, build a decimal calculating machine). Nevertheless, the ideas were all there. The question immediately arises, then, how much this work may have influenced Zuse or any of the other computer pioneers. It would have seemed likely that Babbage would have been aware of Leibnitz's work, for he mentions in his autobiography how he spent his student days at Cambridge trying to promote Leibnitz's notation for the differential calculus over that of Newton. However, there is no reference by Babbage to Leibnitz's writings on combinatorics or binary arithmetic that I have been able to locate. And as I mentioned, his proposed Analytical Engine, though brilliant in design, does not resemble Leibnitz's ideas in its logic. There is the possibility that the controversy between Leibnitz and Newton over the discovery of the differential calculus prevented his ideas from getting a fair hearing in England, but on this point I can only speculate.

We do know that in Germany in the 1920's and 30's, Leibnitz's thoughts on binary arithmetic were mentioned in the **Hochschule** mathematics courses, if only in a casual way. And Herr Zuse was able to draw on this prior work as a guide to his own. I do not mean to say that Herr Zuse built the machine that Leibnitz wanted to build, but I do want to show that there has been a continuity of progress through the years, even if only by a thin thread. I want to reemphasize by this example, however, that it is one of the historian's priorities to establish such links to the past; or, if he should not find one where he might expect it, to try to explain why it was not there. Notice also by this example that not only do we gain a better understanding of the path by which Herr Zuse came to invent the computer, but we also gain a new appreciation of a part of Leibnitz's writings that until now have been pretty much neglected.

In a complementary fashion, we are interested in tracing the influences that Herr Zuse's work had on the later development of computers. But because of the wartime conditions, we find that much of Zuse's early work was done independently of other parallel developments, not only in the United States and England, but also in other cities in Germany. Thus we have in the example of Herr Zuse a striking example of the independent discovery of the same basic design principles of the computer.

Briefly mentioned, these are:

- 1) The discovery and application of the statements of symbolic logic as the basis for the design of the control and arithmetic units of his machines. Implicit in this discovery is the use of the binary system, as well as the realization that statements formulated in symbolic logic could be directly translated into electrical circuits using relays.
- 2) The understanding of the concept of computability in its most general sense -- that one could in theory design a general-purpose machine with a simple structure that could solve any problem that can be programmed into it. (This criterion, in effect, becomes the definition of computability). Herr Zuse's definition of "Rechnen" illustrates this: "From given statements, to produce new statements, according to a prescribed plan."
- 3) Perhaps most important, Herr Zuse set out to build such a universal machine, and was the first to complete a working calculating device which can be said to embody the principle of generality. This was of course, the Z-3 relay machine which was completed in 1941.

The phenomenon of multiple independent discovery, wherever it occurs in the history of science, provides the historian with a powerful tool with which he can analyse the processes of invention, discovery, and technological advancement. He can do this in two ways.

In the first case, we look at those aspects of discovery which were identical in the various places. If this is so, we may then say that what was discovered may lie at the core of our subject -- that it is a part of a larger truth. On the other hand, those aspects which are significantly different may reveal to us the effect of the individual talents, prejudices, and backgrounds of those involved. We can illustrate each of these approaches to history with the example of Dr. Zuse's work.

Consider first Herr Zuse's adoption of the binary system of enumeration. As I have said, this represented a clean break with the prevailing ideas of the calculating machine industry, which at that time was building ever more sophisticated decimal machines. Herr Zuse has recounted the story of how that industry was concerned with the problem of machine multiplication -- they were divided into two camps, the one in favor of multiplying by repeated addition, the other by direct table look-up. Now in binary the multiplication table is simply 4 for 1x1, and 0 everywhere else. Thus binary multiplication is primarily a matter of summing the partial products, so the two methods become identical. So we might say that Herr Zuse cut the Gordian Knot of machine multiplication, rather than trying to untie it. And, just like the legend, people were not sure at first that he had, in fact, solved the problem.

Now the advantages of the binary system were also seen by several others at the time: in France, by Valtat, in England, by Wynn-Williams, and in America by Atanasoff and George Stibitz. It appears that each of these people arrived at the idea sometime in the 1930's, without knowledge of each other's work (Valtat was a little earlier).<sup>8</sup> Thus each deserves about the same credit as to the priority of recognizing the binary system's advantages.

To the historian, the question is not one of establishing the exact chronology of who was the first, although he may wish to do this. The more interesting question is to ask whether this simultaneous discovery indicates something about the fundamental nature of the digital computer. In other words, when we say that the computer was invented at this time, it is because the machines that were built in the 1930's were fundamentally different from just larger calculating machines, or from several business machines linked together, as L.J. Comrie had done in England in the 1920's. This qualitative difference is brought out by the adoption of base-2. It shows us that what we regard as a computer has a logical as well as arithmetical power to it, so much so that it justifies the adoption of a number-base that is unfriendly to human beings. This is not to say that decimal computers are impossible, but it gives the historian a guide through the great mass of evidence about past events.

On the other hand, a closer examination of the work of the men just mentioned reveals important differences, and these differences are also of great interest to the historian today. Whereas I feel that several people should share credit for recognizing the advantages of binary, only Herr Zuse completed a full-scale working machine which performed all of its internal calculations in the binary scale. In the U.S., John Atanasoff began work on a machine with a similar design, but he never went beyond the prototype stage. George Stibitz, a research mathematician with the Bell Telephone Laboratories, also constructed prototype binary circuits, but in the design of the relay computers that Bell Labs constructed throughout the 1940's, Stibitz adopted, not pure binary, but binary coded **decimal** (in the Model I), or other coded decimal representations. Thus Dr. Stibitz's machines, which because of their relay technology most resemble Zuse's externally, had a very different design.<sup>9</sup> The Model I did code each decimal digit by the binary code for 3 greater than each digit, but all internal arithmetic was done in the decimal scale.

In the case of the ENIAC, the first large-scale electronic computer, its designers knew of Wynn-Williams's work in building high-speed binary counters using vacuum-tube technology. However, they rejected this design and chose instead to build a decimal machine in which the tube circuits

functioned as the exact counterparts to the toothed wheels of a decimal mechanical calculator.<sup>10</sup>

These differences reveal that while the concept was very much "in the air" and waiting for someone to realize it, the actual way in which people did recognize and make use of the binary system varied from one individual to another. It shows us that the computer, like any other invention, was the product of individuals, whose personal judgements were necessary at every stage.

Herr Zuse's decision to carry out all internal operations in pure binary indicate his belief that in terms of the total amount of time that a computer would require to solve a problem, the number of internal operations would far outweigh those necessary to convert to and from decimal (at the beginning and the end of the problem). The others apparently did not agree with this judgement, or else they had other reasons to hold on to the decimal representation. We should always keep in mind that in the 1930-s all of this was an uncharted area, and there was no body of theory upon which the designers could rely for guidance. Indeed, it was not until the 1950's that this question was finally resolved, with most large-scale computers using pure binary as Dr. Zuse proposed, and small desk-top electronic calculators using a version of binary-coded-decimal first pioneered by Stibitz.

I would like to examine one more aspect of Dr. Zuse's work which will reveal some of the difficulties that the historian must confront in understanding the past. Besides the use of binary numeration, Herr Zuse's computers were distinct from other calculating machinery in that they had a clear separation of components into those for memory, and those for arithmetic and control. In this respect the Zuse machines most resembled the proposed Analytical Engine of Babbage's, where the latter' "Store" and "Mill" correspond to Zuse's "**Speicherwerk**" and "**Rechenwerk**," Herr Zuse first learned of Babbage only in 1937, after he had applied for an American patent on some of his early designs. As far as I can determine, no other machine, except Zuse's, built before the end of World War II so clearly had this separation of components. In America, the Harvard Mark I machine, which was designed by Howard H. Aiken, was acknowledged as "Babbage's Dream Come True," primarily because Professor Aiken gave credit to Babbage as the inspiration to build such a machine. But this can be misleading. In the Mark I, the bulk of the computing was done in the 72 storage registers, with only multiplication and division being done in a separate unit. The ENIAC also shared this design feature. In these machines, the central component was the **accumulator**, a device which both stored and added numbers. Once again, this type of architecture reveals the influence of the calculating-machine industry, whose mechanical registers functioned in the same way. (The Bell Labs's machines, with their relay construction, did not use accumulators, but they do not show as clear a separation of functions as either Babbage's design or Zuse's Z-1 through Z-4.)

Now it is well-known that today's computers also exhibit this kind of design. It has come to be known as the "von Neumann" type of architecture, because of the enormous influence that von Neumann had on post-war computer activity. There is no evidence that von Neumann derived any of ideas from either Zuse or from Babbage, so we may assume that it was an independent decision.

I should mention here that some historians are reluctant to use the word "computer" when describing the early achievements of Dr. Zuse, since the machines he built at that time did not have either conditional branch facilities or the ability to internally store programs.<sup>13</sup> I hope that by the examples that I have just given it is clear that there is no one or two criteria that we can use to define a computer, either in 1941 or even today. Rather than trying to pin down such a definition, we should try to establish a general picture of each of the early machines, paying attention to the sum of all of each machine's features.

It is true that the Z-3 computer was only capable of executing fixed programs without conditional

branches. On the other hand, the elegance of its architecture meant that such a capability could have been built into it with little difficulty, if the need for it was felt. In fact, this was done with the Z-4, which had conditional branch capabilities built into it after the war. Once again I want to stress the dangers of applying the standards of today to the record of the past. We should realize that while by today's standards the stored-program principle is a **sine qua non** of computers, it may not be so in the future. Just as what we now call programmable pocket calculators would have, by anyone's standard, been called a computer in 1953, so too may the computers of today not satisfy the criteria of tomorrow. Historians are part of this ongoing process, I might add. To be fair, we must also admit that it is not correct to use the fact that the Z-3's architecture resembles today's computers as a measure of its quality. What we must do is compare the Z-3's computing power to that of the other machine of its day. The Z-3 was a much more modest machine than the others, although its memory was somewhat greater. It could not handle problems as complex as those that the Mark I could handle, nor was it as fast as the ENIAC, of course. But it was as reliable as any of the others, was as fast as the other electromechanical computers, and in terms of the amount of computing power for its size, it was more efficient than the others. The extra efficiency was no doubt due to its more elegant architecture.

I would like to turn now to some themes which require more research by historians before we can make any definite conclusions.

The first of these is the extent to which Herr Zuse's work influenced later computer development. Because of the secrecy imposed by the war, there was little chance for news of his work to travel outside Germany until after 1945, and even then it was only gradually and imperfectly. Herr Zuse did manage to salvage one of his machines, the Z-4, after the war's end, but for several years it was only partially functional. When he was visited by Allied Intelligence forces in the Alpine village of Hinterstein, it seems that the Allied investigators failed to grasp the significance of what they saw. Because only now are some of the security restrictions on this information being lifted, there is opportunity for much further research here.<sup>15</sup> The result of these events was that the many projects, conferences, and discussions on computing that were taking place in the U.S. and England immediately after the war did not benefit from Dr. Zuse's possible contributions. Therefore we must conclude that the immediate ancestor of today's computer was the ENIAC, despite its backwardness of design already mentioned. Even as the ENIAC was nearing completion, its builders recognized the drawbacks of these features and made plans for future machines that would be more compact and efficient.

The path from the first program-controlled machines to the modern computer is not within the scope of this address, but the few details just mentioned should show how far it was from being straight and smooth. Even where there was an open exchange of ideas after the war the process of trial and error continued. It also should show that what we often call "milestones" of computer history have turned out to be "millstones," in that their very success made it difficult for people to see that an alternative approach might be better. In later years, as computer design evolves, people look back to the past and find new "milestones," in machines that were the first to exhibit some property. In any case, for external reasons, the work of Herr Zuse did not have the chance to be as influential as it should have been.

### **External Factors**

Thus far we have been discussing the design features of the early machine in an abstract way. But what I have just called "external" factors also played a significant role. Much has been written about the emergence of the modern computer industry after the war, and how the governmental policies of America, Germany, and England each affected that industry in their respective countries. We now live in an era of "Big Science," where government agencies give out large sums

of money in hopes of realizing an even larger economic return. Naturally such agencies, and anyone else who is a part of this society, would be most interested in any historical evidence that would show which kinds of policies were most effective in the past. I must warn my audience at the start that the historian can only make tentative statements about these matters, but at the same time,

I do not rule out the possibility that in the future such research will prove to be much more valuable.

Alfred North Whitehead made the remark about the age in which he lived that the greatest invention of that age was the invention of the method of invention itself -- that by the systematic application of scientific research one could generate technological progress continuously. We may use the example of Konrad Zuse's construction of his computers to illustrate and qualify this statement.

We have already seen that Herr Zuse came to build a computing machine independently of both the calculating machine industry and the scientific instruments manufacturers. He began pursuing this goal with only the support of a few close friends, and he worked in the converted living room of his parents' residence in Berlin. Later on he began to get help from the German Research Institute for Aircraft (DVL), but by this time he had already formulated the basics of the design and method of construction for his machines.

I think that this example illustrates that the best that we can hope for is for governments to make it easy to recognize novel solutions to problems, but I do not see how such agencies can anticipate a breakthrough. The earliest stages of discovery are the most critical, but it is precisely here where we have few standards by which to recognize its value.

Those who supported Herr Zuse during those early stages, either with small sums of money or by helping cut and assemble the parts of the Z-1, did so because of their personal faith in their colleague's vision, not because they necessarily understood the significance that such a computer would have for the future. So to return to Whitehead's statement, the rise of research institutes and laboratories has made it easier perhaps for innovative ideas to be recognized and supported, but they do not necessarily affect the **source** of such ideas.

In this context we can get a better idea of the role of the Second World War on the invention of the computer. It has often been remarked that the computer could have been invented much earlier, but it was only the pressures of the war that gave the necessary impetus to actually produce them. We should note, however, that all of the basic ideas for computers were formulated before the war, and furthermore, many people had begun to build prototype machines. In some cases, like the work of Atanasoff and of Helmut Schreyer (who was one of Zuse's co-workers), the pressures of war-time activity put an end to their prototype development. Had there not been the war, there most certainly would have been computing machines, probably they would not have turned out so large as they did. Certainly they would have been better designed, since people would have had more time to think through the results of their efforts. But they would have been built. We can only speculate, but had it not been for the(y) war there might have been more advanced computers than the numerical specification of these positions even sooner. I can cite as evidence for this belief the fact that Herr Zuse has often mentioned how it was the desire by the DVL for a machine that was suited for specific wing-flutter calculations that held him to a restricted design for the Z-3 and Z-4 machines.<sup>16</sup>

As to the question of understanding the very sources of invention and discovery, we have even less solid evidence on which to base our judgements. About all that we can say is that great ideas seem to come from people who can apply the skills learned in one problem-solving area to the problems

faced in another, quite different area. Thus it appears that the British code-breaking machines were built, not by cryptanalysts, but by mathematicians who also were exceptionally good chess-players.<sup>17</sup> Apparently they could transfer their skill at solving chess problems to the problem of decoding secret messages.

In the case of Herr Zuse's work, I would venture to suggest that it has been his talent for drawing and for the graphic arts that has helped him make breakthroughs in machine computation. Today we are accustomed to think of the engineer as living in a world far different from that of the visual artist, but Herr Zuse has recounted how he was equally attracted to both worlds as a student.<sup>18</sup> Indeed the work of Herr Zuse is characterized throughout by a graphical approach. His first line of investigation was to design a 2-dimensional pattern into which numerical quantities would be placed. Their position in the pattern would determine whether they were to be multiplied, added, etc., to each other. Later on he saw that the numerical specifications of these positions in space would make their actual positioning unnecessary. It is interesting that he chose the word "Plan" to describe this numerical information. This word has a graphical connotation (in both German and English) that the word "program" does not. His later development of the symbolic language which he called "Plankalkül" also had a 2-dimensional format, and his most recent work on "Rechner Raum" shows that he continues to look at world of computation with the eye of the visual art as well as computer scientist.<sup>19</sup>

Once again we must admit that this particular aspect of Herr Zuse's life may be just coincidental to his scientific and technical achievements, but it is certainly one of the themes that the historian should deal with. Perhaps someday this process of fortuitously combining skills in different subjects will be understood to the point where it can be systematically applied, perhaps not. In any case, these questions should be at least seriously asked.

## Conclusion

I would like to conclude this address by saying that I have only given a personal and incomplete account of the early work of Herr Zuse. There are many aspects of that history that I have neglected to mention, and I realize that other historians might have presented a different selection of the oast than what I have. The historian is always faced with the task of selecting out what he feels were the main issues of the past, of course always remembering that what he selects must be fair to the facts that he has gathered. But it is not possible to separate the fact-gathering from the inter-pretation. Because the history of computing is such a young field, we naturally must spend most of our time in gathering data, interviewing the participants, assembling artifacts and photographs, etc. This is especially true of Herr Zuse's work, since so much of it was lost in the war that what has survived becomes all the more important. But if we are to make sense of all these facts we must be willing to make some rather bold speculations, even when based on fragmentary evidence. As an American student of the history of what I had at first thought to be an American invention, I have come to realize that we must rewrite and revise the story of the invention of the computer. But I must also emphasize that this process of revision and reinterpretation is an ongoing process. I hope that I have conveyed some of the felling of this process in this address.

## References

- 1) George Sarton, *The Study of the History of Science*, (New York, 1936), p. 5.
- 2) Herbert Butterfield, *The Whig Interpretation of History*, (New York, w,w. Norton Co., 1965).
- 3) Konrad Zuse, *Der Computer; Mein Lebenswerk*, (Verlag Moderne Industrie, 1970), p. 36.
- 4) see, for example, Friedrich A. Willers, *Mathematische Instrumente*, Berlin & Leipzig, 1926 (Sammlung Göschen, Bd. 922); also Meyer zur Cappellen, W., *Mathematische Instrumente*, 3.

Auflage, Leipzig, 1949.

5) Thomas Parke Hughes, "ENIAC: Invention of a Computer," *Technikgeschichte* 42 (1975), pp. 148-165.

6) Knuth, Donald E.: "Ancient Babylonian Algorithms," Association for Computing Machinery, *Communications* 15(1972)» PP. 671-77; 19(1976), p. 108.

7) Herr Buttman's letter has been reproduced in Karl-Heinz Czauderna, "Konrad Zuse - Der Weg zu Seinem Computer Z-3 und dessen Verwirklichung," Abschlussarbeit, Fachbereich Elektrotechnik, Fachhochschule Augsburg, Jan, 1978, in Appendix.

In one of Babbage's essays he mentions his desire as a student to learn about the possibilities of a Universal Language. In the very next paragraph he mentions his reading of the works of Leibnitz on the differential calculus, but he does not make any mention of Leibnitz's similar interest in such a language: Charles Babbage, *Passages From the Life of a Philosopher*, ed. by Philip & Emily Morrison, (New York 1961), p. 22.

8) Raymond Valtat, 'Machine a calculer fondee sur l'emploi de la numeration binaire. *Comptes Rendus Acad, Sci, Paris* 202(1936) pp, 1745-47, C.E. Wynn-Williams, "A Thyatron 'Scale of Two' Automatic Counter," Royal Society of London, *Proceedings A* 136 (1932), pp. 312-24.

George Stibitz & Evelyn Loveday, "The Relay Computers at Bell Labs," *Datamation* 13 (April 1967), pp. 35-44, May, 1967, pp. 45-49.

9) Information on the Bell Labs Computers has been taken from the collection of the papers of George Stibitz at the Baker Library, Dartmouth College, taken from the Stibitz at the Hanover, New Hampshire.

10) J. Presper Eckert, "A Survey of Digital Memory Systems," Institute of Radio Engineers, *Proceedings*, 41 (1953), P. 1394.

11) Konrad Zuse, "Method for Automatic Execution of Calculations with the aid of Computers," German Patent application Z 23 139 IX/&2m, 11 April, 1936. Reprinted in Brian Randell, ed., *The Origins of Digital Computers, Selected Papers*, (Berlin, Heidelberg & New York, 1975, second ed.) p, 163.

12) Zuse, *Der Computer, Mein Lebenswerk*, p. 149.

13) See, for example, Fred Gruenberger, *Datamation* (Nov. 1977) P. 24.

14) Konrad Zuse, "Some Remarks on the History of Computing in Germany," paper presented at the International Research Conference on the History of Computing, Los Alamos, New Mexico, June 10-15, 1976, plate 8.

15) British Intelligence Objectives Subcommittee, "Applied Mathematics in Germany, with Particular Reference to Naval Applications," B.I.O.S. Report No, 79 [1947], pp. 63-4. Also R.C. Lyndon, "The Zuse Computer," *Mathematical Tables and Other Aids to Computation*, 2 no. 20 (1947), pp. 355-359. Lyndon was not permitted to examine the entire Z-4 computer, which at the time was in the village of Hinterstein, in the Allgäuer Alps.

**Konrad Zuse -  
Konstrukteur der ersten programmgesteuerten Rechenanlage**

Prof. Dr. Eike Jessen

Im Jahre 1941 nahm die erste funktionsfähige programmgesteuerte Rechenanlage, Konrad Zuses Z3, ihren Betrieb auf. Seitdem sind 38 Jahre vergangen. Die Dynamik, mit der in dieser kurzen Spanne die programmierte Datenverarbeitung sich verbreitet hat, durfte jedem eine Ahnung von der Wirksamkeit, ja von der Bedrohlichkeit dieses technischen Mittels geben.

Erstaunlicherweise wurde diese für uns heute so grundlegende Technik nicht vorhergesehen. Eine Prognose, die namhafte Wissenschaftler 1937 über den technisch-wissenschaftlichen Fortschritt bis zum Jahre 1967 machten, erwähnt die Möglichkeiten automatischer Datenverarbeitung mit keinem Wort.

Um so höher ist die erfinderische Leistung der Pioniere der Datenverarbeitung einzuschätzen, insbesondere die Leistung Konrad Zuses. Als Zuse seine Entwurfsarbeiten 1933/34 aufnahm, gab es wohl eine Büromaschinenindustrie, die Tisch-Rechenmaschinen in Jahrhunderten zur Reife entwickelt hatte und ~ als jüngeres Produkt - leistungsfähige elektromechanische Lochkartengeräte hervorgebracht hatte, Produkte, die nur ganz eingeschränkte Verwandtschaft zur programmgesteuerten Rechenanlage haben. Mit ihnen lassen sich zwar elementare Rechenoperationen, nicht aber komplexe Rechengänge automatisch erledigen. Zuse selbst lernte die sich hieraus ergebenden Mängel in seinem Beruf als Statiker kennen. Von hier aus gewann er den Anstoß zu seiner Rechnerentwicklung, die über die Maschinen Z1 und Z2 zur genannten Z3 führte.

Zuses entscheidende Idee war, eine universelle Maschine zu bauen mit einem Rechenwerk für die arithmetischen Elementaroperationen und einem Speicher mit nummerierten Zellen. Er sollte dazu dienen, Werte, die von einer Tastatur oder vom Rechenwerk eingegeben wurden, auf Abruf dem Rechenwerk zu übermitteln. Ergebnisse wurden auf einer Anzeige dargestellt. Die gesamte Maschine arbeitete schrittweise gesteuert von einer Befehlsfolge, die sie von einem gelochten Filmstreifen übernahm. Das heißt: eine Maschine zu bauen, die aus Daten nach Maßgabe von Daten neue Daten erzeugt. Unschwer sind bereits in Zuses erstem Konzept die vier großen Funktionseinheiten Rechenwerk, Speicher, Ein/Ausgabe und Leitwerk zu erkennen, die bis heute in jeder Rechenanlage zu finden sind.

Zuse erkannte, daß eine so komplexe Maschine nur dann konstruiert werden und arbeiten konnte, wenn ihre Operationen auf elementarste und sicherste Funktionen gegründet werden konnten. Er wählte deshalb das Binärprinzip und entschied sich für Dualzahlen, wofür es in den am Markt angebotenen Rechenmaschinen kein Beispiel gab, obwohl schon Leibniz das Binärprinzip unter anderem im Hinblick auf die Mechanisierung des Rechnens untersucht hatte.

Es ist besonders kennzeichnend, daß es die überlegenen technischen Eigenschaften dieses Prinzips waren, die Zuse sich so entscheiden ließen, nicht etwa die Möglichkeit, Relais als Bauelemente einzusetzen. Diese schienen ihm für eine so komplexe Maschine zunächst wenig geeignet: Platzbedarf, mangelnde Zuverlässigkeit und nicht zuletzt nicht tragbare Kosten. Zuse entwickelte vielmehr eine reine mechanische Schalttechnik, die er - dank des Binärprinzips - homogen für Steuer-, Speicher- und Verarbeitungsfunktionen einsetzen konnte.

Auf dieser technischen Basis gelang Zuse 1936 der Bau eines mechanischen Speichers von 1,3 KBit für seine erste Maschine, die Z1. Die der Binärtechnik angeborenen Sicherheitsvorteile erlaubten erstaunlicherweise die Herstellung der Schaltteile mit einer Laubsäge. Die Technik bewährte sich so gut, daß noch 1944 das Gerät Z4 mit einem ähnlichen Speicher ausgestattet werden konnte.

Bei der Begründung seines Konzepts verließ Zuse noch in einem weiteren wesentlichen Punkt das Vorbild der konventionellen Rechenmaschinen. Zuses Rechenanlagen speicherten und verarbeiteten von Anfang an Zahlen in halblogarithmischer Darstellung. Wird ein Rechengang automatisiert, dann kann kein Betrachter darüber wachen, daß die Werte im Darstellungsbereich der Maschine bleiben; bei konventionellen Rechenmaschinen korrigiert der Benutzer den Dezimalpunkt in geeigneter Form, um die Genauigkeit der Maschine voll nutzen zu können. Die halblogarithmische Darstellung erlaubt, automatisch in einem stark erweiterten Bereich mit annähernd gleichbleibender Genauigkeit zu rechnen, und erübrigt so meist die Einsicht und Einwirkung des Benutzers.

Die Anlage Z1 erwies sich als noch nicht tauglich. Ein neuer Ansatz, die Z2, erhielt zum Speicherwerk der Z1 nun ein Relaisrechenwerk. Diese Maschine war zwar für einen praktischen Betrieb noch nicht verwendbar, demonstrierte aber immerhin die Richtigkeit von Zuses Prinzip. Zuse konnte diese Maschine der Deutschen Versuchsanstalt für Luftfahrt vorführen.

So lange hatte Zuse seine Arbeiten aus eigenen Mitteln und denen von Freunden und Förderern bestritten. Mit der geglückten Demonstration der Z2 gewann er die finanzielle Unterstützung der Deutschen Versuchsanstalt für Luftfahrt für den Bau einer technisch verbesserten Maschine, der bereits anfangs genannten Z3, der ersten voll funktionsfähigen programmierbaren Rechenanlage. Sie wurde als Relaismaschine gebaut. Von den 2600 Relais wurden allein 1400 für den Speicher, wiederum 64 Worte zu 22 Bits, aufgewendet. In heutiger Sprechweise war die Z3 eine Ein-Adreß-Parallelmaschine mit externer Programmsteuerung über 8-Kanal-Lochfilm, zehn Befehlen, 22 Bits Datenwortlänge (Gleitpunkt), 64 Worten Speicherkapazität, Gleitpunktrechenwerk, Dateneingabe über Tastatur und Lampenfeldanzeige als Ausgabe. Die Maschine leistete im Mittel etwa eine Operation je Sekunde. Unter den Operationen war auch Multiplikation, Division und sogar das Radizieren. Die Z3 ging 1941 in den praktischen Betrieb für die Deutsche Versuchsanstalt für Luftfahrt und wurde mit Erfolg u.a. zur Berechnung von Eigenschwingungen (Flutterfrequenzen) von Tragflächen eingesetzt.

An dieser Stelle muß etwas über das technische Niveau von Zuses bahnbrechender Entwicklung gesagt werden. Bereits mit seinem ersten Relaisrechner erreichte Zuse nämlich einen beispielhaften technischen und methodischen Stand. Anders als in den Vorbildern für große Relaissysteme, den Fernsprechvermittlungssystemen, benutzte Zuse für seine Maschinen einen zentralen Taktgeber. Damit erreichte er, daß alle Zustandsänderungen in der Maschine synchron und nach zentralem Zeitmaßstab vollzogen wurden (und zeitlich nach Maßgabe des Taktes), seit Zuses Erfindung ein typisches Kennzeichen von Rechnerschaltwerken. Dazu verließ Zuse die in der Vermittlungstechnik üblichen Relaischaltungen mit Eigenzeitkonstanten und gewann dafür den Vorteil, daß er seine Maschine im Einzelschritt testen konnte. Er sah eine zentrale Schaltwalze vor, die es ihm erlaubte, die Relaiskontakte stromlos zu schalten, was entscheidend für die Lebensdauer eines großen Relaissystems ist.

Die wichtigste Qualität des Ingenieurs Zuse war aber die Verbindung von konzeptioneller Weitsicht und technologischem Geschick mit Abstraktionskraft: Diese Fähigkeit ließ ihn nicht nur einen logischen Kalkül für den Schaltungsentwurf entwickeln, sondern auch seinen Plankalkül für die Entwicklung von Programmen, lange vor den ersten technischen Programmiersprachen. Hier sah er auch die Möglichkeit vor, den Ablauf einer Berechnung von ihren Ergebnissen abhängig zu machen, eine Möglichkeit, die es in der Z1 bis Z3 nicht gab.

Zuse hatte diesen folgenschweren Schritt schon 1937 vorausgesehen und notiert. Für die Aufgaben der Z1 bis Z3 war er nicht zwingend. Technisch hatte er das programmgesteuerte Vor- und Zurücksetzen des Befehlslochstreifens erfordert, besser aber die interne Speicherung nicht nur der Werte, sondern auch des Befehls in der Maschine. Daran war aber nach dem Stand der Speichertechnik zunächst nicht zu denken. Als 1945 von Neumann die interne Programmspeicherung und damit sogar die Selbst-Modifikation des ablaufenden Programms forderte, geschah das in der Hoffnung auf Ladungsspeicherröhren als billige und schnelle Speicherelemente, eine Technik übrigens, die die Hoffnungen nicht erfüllte.

Mit der Z1, Z2 und Z3 sind Konrad Zuses grundlegende Beiträge zur Organisation digitaler Rechenanlagen nicht erschöpft. Er baute noch während des zweiten Weltkrieges einen Spezialrechner S1 mit starrem Programm zur Korrektur der Herstellungsauigkeiten von Flugkörpern. Zuse rüstete ihn später mit Meßuhren und Analog-Digital-Wandlern aus und schuf damit die erste direkt gekoppelte Meßwertverarbeitungsanlage. 1944 begann er mit der Projektierung einer Rechenmaschinengeneration nach der Z1 - Z4-Klasse und sah dort außer Programmspeicherung auch assoziative Speicher vor, also Speicher, deren Zellen nicht durch eine Adresse sondern aufgrund von Inhaltskriterien zum Lesen oder Schreiben ausgewählt werden. 1957 untersuchte Zuse Rechenprobleme, in denen eine große Zahl von Daten gleichzeitig denselben Operationen zu unterziehen ist, und entwarf dazu eine Maschine, die er Feldrechenmaschine nannte. Nach Zuses Entwurf waren die Datensätze auf einer Trommeloberfläche angeordnet und wurden seriell von einem Rechenwerk bearbeitet. Zuse hat damit den ersten SIMD-Rechner (single instruction - multiple data streams) angegeben. Assoziative Speicher und SIMD-Rechner werden erst durch Technologien dieses Jahrzehnts attraktiv; Verfahren zu ihrem wirkungsvollen Einsatz werden Anwender und Informatiker noch lange beschäftigen. - Spätere Arbeiten von Konrad Zuse haben der Vereinigung quantenphysikalischer und automatentheoretischer Sichtweisen gegolten sowie der Untersuchung der Eigenschaften und Möglichkeiten selbstreproduzierender Maschinen.

Die Informatik ist eine sehr junge Wissenschaft. Den praktischen Anforderungen aus der Datenverarbeitung kann sie in wichtigen Punkten noch keine gesicherten Lösungen entgegenstellen. Es beim Abstrahieren, Erwägen und Gliedern zu belassen, liegt leider oftmals nahe. Aber die Informatik braucht die technische Durchsetzung ihrer Ergebnisse. Und Konrad Zuse steht nicht nur zeitlich als einer der Pioniere vor uns, er hat auch gehandelt. Er hat gebaut und durch seine Maschinen den Wert seines Konzeptes bewiesen. Genauso selbstverständlich hat er später die Wagnisse eines Unternehmers auf sich genommen. Die Zukunft unseres Landes hängt unmittelbar von unserem technischen Vermögen ab. Als Informatiker und Bürger dieses Landes sehen wir in Stolz und Dankbarkeit auf Konrad Zuse.

**Formulierung, Formalisierung und automatische Programmierung  
in den frühen Arbeiten Konrad Zuses**

F. L. Bauer

Festvortrag, gehalten anlässlich der Verleihung der Würde  
einers Dr. rer. nat. h.c. an K. Zuse, Hamburg, 10.10.1979

Konrad Zuse ist als Pionier der modernen Rechenmaschinen bekannt. Seine technischen Leistungen haben gebührende Beachtung gefunden. Seinen Zeitgenossen Aiken, Stibitz, Eckert und Mauchly ist er nicht nur zeitlich eine Nasenlänge voraus, er übertrifft sie auch in der Reichhaltigkeit und Tiefe der Ideen. Bei ihm finden sich von Anfang an die konsequente Verwendung des Dualsystems, die Einführung der "halblogarithmischen Form" von Zahlen als Basis der Gleitpunktrechnung, die Verwendung eines binären Codes für Wählpyramiden, die über eine starre Wiederholung hinausgehende, bedingungsabhängige Programmsteuerung, wie sie im Plankalkül unterstellt war und in die Züricher Z4 einging. Bezeichnend ist die Erkenntnis, daß mechanisiertes Rechnen allgemeinste Zeichenverarbeitung umfaßt.

Diese funktionelle Seite Zuses hat mich von jeher interessiert, ihr will ich in diesem Vortrag nachgehen. Zuse meint wohl sie, wenn er ([1], S. 35), vom "faustischen Geist" in ihm spricht.

Formulierung

Nach seinem eigenen Zeugnis ([1], S.35) war Zuses Ausgangspunkt das Arbeiten mit Formblättern und Rechen-Schemata, konkret zur Durchführung statisch unbestimmter Rechnungen. Die Formel, nach der zu rechnen ist, schlägt sich im Schema nieder. Die Aufstellung geeigneter Rechenformulare gehörte damals zu den Standardfähigkeiten eines angewandten Mathematikers. Ein Beispiel (für die Auflösung linearer Gleichungen durch Gauß-Elimination) findet sich bei WILLERS 1928([2]), siehe Abb. 1.

	$x_1$	$x_2$	$s$	
I	$a_{11}$	$a_{12}$	$a_{13}$	$s_1$
II	$a_{21}$	$a_{22}$	$a_{23}$	$s_2$
la		$\frac{a_{21}}{a_{11}} a_{12}$	$\frac{a_{21}}{a_{11}} a_{13}$	$\frac{a_{21}}{a_{11}} s_1$
III	$a_{31}$	$a_{32}$	$a_{33}$	$s_3$
lb		$\frac{a_{31}}{a_{11}} a_{12}$	$\frac{a_{31}}{a_{11}} a_{13}$	$\frac{a_{31}}{a_{11}} s_1$
IV	$\sigma_1$	$\sigma_2$	$\sigma_3$	$\Sigma$
lc		$\frac{\sigma_1}{a_{11}} a_{12}$	$\frac{\sigma_1}{a_{11}} a_{13}$	$\frac{\sigma_1}{a_{11}} s_1$
V = II - Ia		$a'_{22}$	$a'_{23}$	$s'_2$
VI = III - Ib		$a'_{32}$	$a'_{33}$	$s'_3$
Va			$\frac{a'_{32}}{a'_{22}} a'_{23}$	$\frac{a'_{32}}{a'_{22}} s'_2$
VII = IV - Ic		$\sigma'_2$	$\sigma'_3$	$\Sigma'$
Vb			$\frac{\sigma'_2}{a'_{22}} a'_{23}$	$\frac{\sigma'_2}{a'_{22}} s'_2$
VIII = VI - Va			$a''_{33}$	$s''_3$
IX = VII - Vb			$\sigma''_3$	$\Sigma''$

Abb. 1: Rechenformular zur Auflösung linearer Gleichungssysteme.

Zuse gibt in [1], S.36 ein Beispiel an (Abb.2). In Abb. 3 ist dieses Formular mit etwas Kommentar versehen.

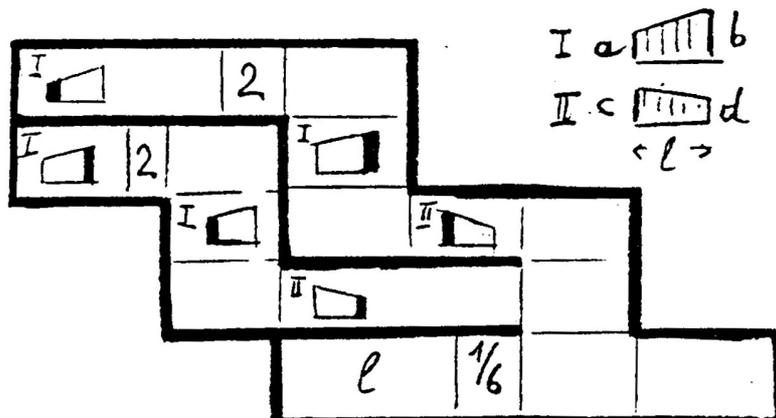


Abb. 2: Rechenformular "zur Berechnung der Übertragung zweier recht-eckig verlaufender Momente". (Von K. Zuse revidierte Fassung.)

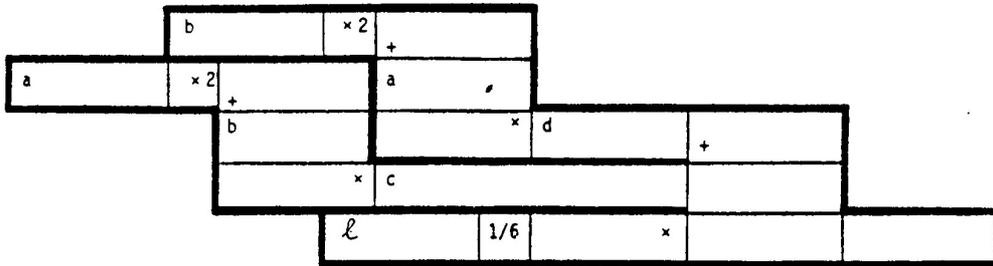


Abb. 3 : Rechenformular für  $l \times 1/6 \times ((a \times 2 + b) \times c + (b \times 2 + a) \times d)$

(Beachte:  $(2a + b)c + (a + 2b)d = (2c + d)a + (c + 2d)b$  )

Nun sind aber solche Rechenformulare topologisch äquivalent zum Kantorovic-Baum der betreffenden Formel, für Zuses Fall Abb. 4a. Formularrechnen bedeutet also: Rechnen längs des Kantorovic-Baums als "Datenflußplan".

Man beachte, daß ein Datenflußplan allgemeiner als ein Befehlsablaufdiagramm ist: über eine strikt sequentielle Reihenfolge, in der die Operationen auszuführen sind, ist nicht entschieden. Das Diagramm kann in verschiedener Weise ausgefüllt werden, es können auch mehrere Personen nebeneinander daran arbeiten (Parallel-Arbeit). Eine Operation kann allerdings erst dann erfolgen, wenn alle ihre Operanden berechnet sind. Diese natürliche Einschränkung (ich habe sie anderswo "natürliche Sequentialisierung" genannt) müssen alle strikten Sequentialisierungen beachten; man erhält (sequentielle) Programme durch einen Prozeß, der (recht irreführenderweise) als "topologisches Sortieren" bezeichnet wird. Ein mögliches Ergebnis zeigt Abb. 4b - dieses spezielle Aufbrechen der Formeln führt zur typisch kellerartigen Verwendung der Zwischenergebnisse (Samelson und Bauer 1957) nach der Devise: Die Operationen werden (von links nach rechts gesehen) zurückgestellt solange nötig und ausgeführt sobald möglich.

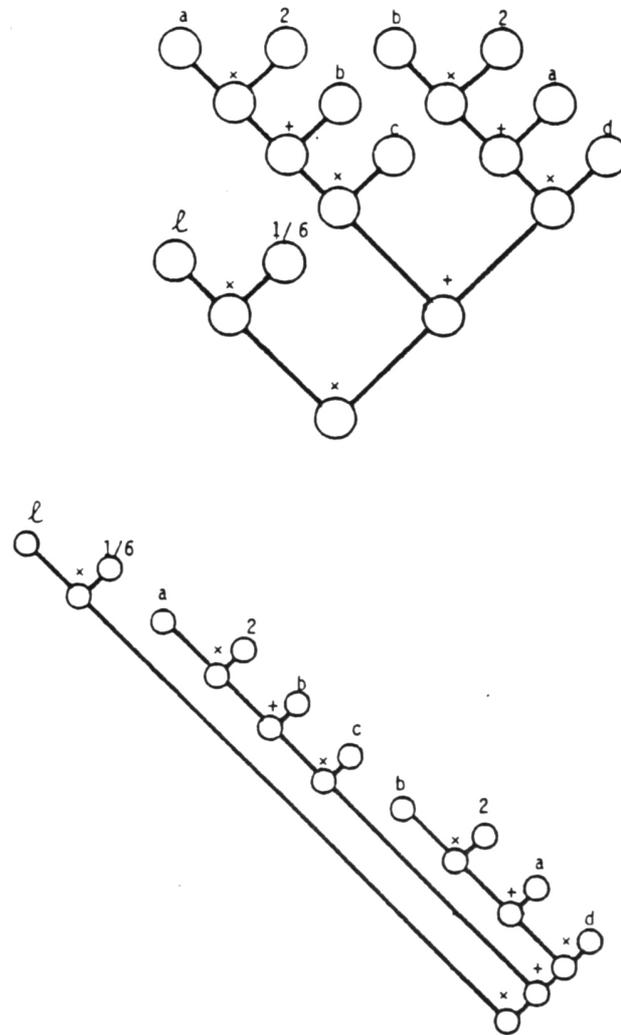


Abb. 4a, b : Kantorovic-Baum zur Formel von Abb. 3 und sequentialisierte Fassung, die durch "topologisches Sortieren" entsteht.

Von solchen Formularen, also Datenflußplänen, ging Zuse aus. Ich kann hier zitieren([1], S. 36):

#### Vom Formular zur Programmsteuerung

Zunächst ging ich etwas kindlich-primitiv vor: Ich dachte mir, daß auf den Formularen F anstelle der von Hand geschriebenen Zahlen an den gleichen Plätzen die Werte eingelocht wurden (Abb. 5).

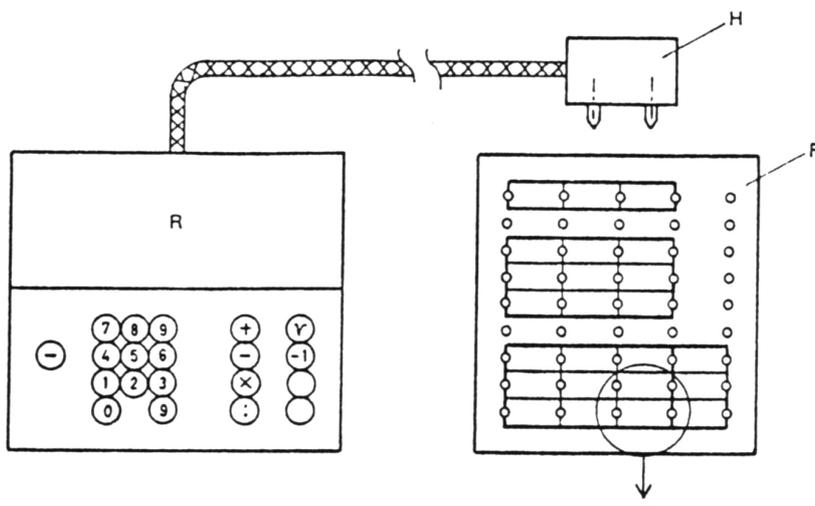


Abb. 5: Versuch zur Mechanisierung der Rechnungen anhand von Formularen, in welche die Werte nicht eingeschrieben, sondern eingelocht werden. Eine vollautomatische Rechenmaschine R kann über eine Abfühl- und Locheinrichtung H von Hand mit den einzelnen Feldern des Formulars verbunden werden.

Die Abtastung und Lochung der Zahlen sollte durch ein bewegliches Gerät H erfolgen, welches von Hand an die betreffenden Stellen gebracht wird und über ein Kabel mit einem Rechenggerät R verbunden ist. Ein erfahrener Fachmann hätte sofort gewußt, daß das nicht funktionieren kann, und hätte vielleicht ganz aufgegeben. Als jugendlich Unbekümmerter aber arbeitete ich weiter. Anstelle der Lochkartenformulare F, die nur einmal benutzt werden können, wollte ich Register Rg setzen, bei denen die Zahlen auf verriegelbaren Stiften S gespeichert werden.

Diese Register können immer wieder gelöscht und neu eingestellt werden. Anstelle der Loch- und Abfühleinrichtungen für Lochkarten würden dann entsprechende Einstelleinrichtungen und Abfühlorgane für die Register treten. Als nächster Schritt müßte die Zuführung dieser Einrichtungen an den richtigen Stellen mechanisiert werden. Dies wäre z.B. durch eine bewegliche Brücke B mit einer Laufkatze K möglich, die das ganze Feld der Berechnungsformulare überstreichen. Dadurch ist garantiert, daß die Loch- und Abfühlgeräte sauber in die richtige Position gebracht werden können, was zunächst noch von Hand erfolgen sollte.

Von einer solchen Vorrichtung ausgehend ist nun ein weiterer wesentlicher Schritt möglich: Die Bewegung der Brücken und Laufkatzen wird durch eine Steuerungseinrichtung St mechanisiert, so daß sie nach einer Vorschrift in die jeweils erforderliche Position gebracht werden können.

Zuse sah nun, daß es sich nicht lohnte, für eine feste Formel ein solches Gerät zu bauen. Es müßte stattdessen ein Gerät verwendet werden, das ein "universelles Superformular" ([1], S. 38) zu bearbeiten erlaubt.

Sieht man einmal davon ab, daß der Gedanke der Rekursivität fehlte bzw. auf Wiederholungen beschränkt war, so zeigen diese frühen Gedankengänge von 1933 ([1], S. 42) heute, 46 Jahre später, bestürzende Aktualität: Der Datenflußplan, der Kantorovic-Baum, ist das Rückgrat der heute so gepriesenen funktionalen, applikativen Programmierung.

Technische Notwendigkeiten, insbesondere der hohe Aufwand, der für Speicherglieder erforderlich war, diktierten im weiteren einen Weg, der von der applikativen Ebene mit der Möglichkeit der Parallelarbeit wegführte. Neben strikte Sequentialisierung des Ablaufs trat der Verzicht auf eine dem Kantorovic-Baum entsprechende Gliederung des Speichers, stattdessen wurde ein homogener, adressierbarer Speicher eingeführt (Abb. 6). Ein Programm für eine solche Maschine mußte erst aus der Formel hergeleitet werden. An die Stelle des "universellen Superformulars" trat der "universelle Superplan", der dies leistete. Wir werden darauf zurückkommen.

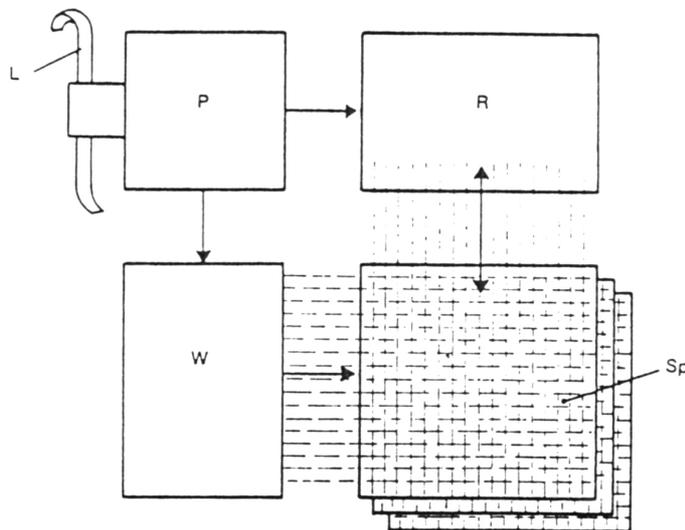


Abb. 6: Erster Entwurf einer programmgesteuerten Rechenmaschine mit einem Programmwerk P, einem Rechenwerk R, einem Speicherwerk Sp und einem Wählwerk W. Anstelle der beweglichen Brücke B (s. o.) ist eine mehrschichtige, mechanische Konstruktion getreten. Die Speicherung erfolgt in Speicherzellen, welche je eine komplette Zahl aufnehmen können. Die Auswahl der Zellen erfolgt über das Wählwerk W. Das Programm wird über einen Lochstreifen L eingegeben, das Programmwerk P gibt die Nummer der Speicherzelle an das Wählwerk W und die Art der Rechnungsoperation an das Rechenwerk R.

Hätte Zuse jedoch noch die Rekursivität zugelassen, so wäre er vom Formular zu einer rekursiven Formularmaschine (Abb. 7) gelangt. Diese wäre einer HERBRAND - KLEENE - Textersetzungsmaschine mit einer Berechnungsregel, die von innen nach außen arbeitet, äquivalent gewesen. Speziell mit dem oben erwähnten kellerartigen Arbeiten hätte sich ergeben, was heute "leftmost-innermost"-Regel genannt wird.

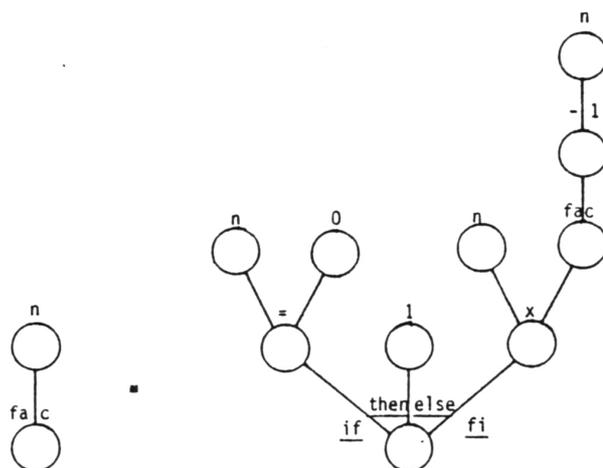


Abb. 7 : Rekursives Formular zur Berechnung der Fakultät.  
(Es wird angenommen, daß genügend viele Exemplare  
des Formulars verfügbar sind.)

### Formalisierung

Ging Zuse beim gedanklichen Entwurf seiner Rechengeräte von Formeln aus, so zeigte sich auch bei der Beschreibung seiner Schaltkreise der Hang zum Formelhaften. Wie er auf die Binärdarstellung von Zahlen stieß ([1], S. 41) hat er in der Literatur nicht verraten\*). Sobald er aber intuitiv (um 1936)

\*) Nach einer privaten Mitteilung war ihm die Binärdarstellung "wahrscheinlich schon aus der Schule bekannt"; die Rechenregeln dazu hat er sich selbst erarbeitet. Der Ausdruck "Sekundalsystem" wurde durch den Prüfer im Deutschen Patentamt geprägt.

den Zusammenhang zwischen binären Schaltkreisen und zweiwertiger Logik verstanden hatte, benutzte er für die Beschreibung von Schaltkreisen Formeln, äquivalent Booleschen Aussageverknüpfungen. Einem damals weitverbreitetem Hang zur Verwendung von Klammern als Operationszeichen (Skalarprodukt, Vektorprodukt) folgend, schrieb er

- (A, B) für die Disjunktion, die er "Verbindung hinreichender Bedingungen A, B" nannte und
- [A, B] für die "Verbindung notwendiger Bedingungen A, B", die Konjunktion, sowie
- A für die Negation von A (vgl. Abb. 8).

So bedeutet etwa [1], S. 64

$$([V_x, V_y] , [-V_x, -V_y])$$

die Formel für das Vorzeichen des Produkts oder den Quotienten in Abhängigkeit von den Vorzeichen  $V_x, V_y$  der Operanden.

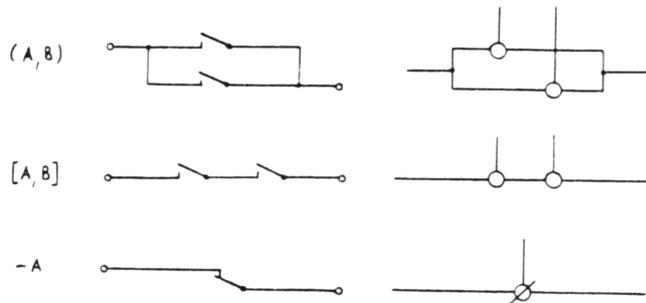


Abb. 8

Zuse beschrieb seine Schaltungen in dieser abstrakten Weise. Er löste sich bereits beim Entwurf seiner Maschine von der verwendeten Technologie und war damit in der Lage, sowohl beim Übergang von mechanischen Schaltgliedern zu elektromagnetischen Schaltgliedern (Übergang Z1 → Z2 → Z3), wie auch später beim Versuchsaufbau elektronischer Schaltungen (um 1938) die Entwürfe unverändert zu übernehmen. Auch die Schaltpläne waren in dieser abstrakten, realisierungsunabhängigen Weise abgefaßt.



### Automatische Programmierung

Programmieren bedeutete das Aufbrechen von Formeln in Abläufe. Technische Zwänge hatten Zuse dazu geführt, Maschinen zu bauen, die programmiert werden mußten. In der Abgeschlossenheit des Allgäuer Dorfes, in das ihn das Kriegsende verschlug, kamen die alten Ideen wieder zum Vorschein. Dort entstand der Plankalkül, eine auch nach heutigen Maßstäben hochentwickelte algorithmische Sprache. Zu einem guten Teil zeigt sie Züge einer Programmiersprache, die auf sequentielle Abläufe eingestellt ist. Andere Bestandteile, wie der Kennzeichnungsoperator, sind bezeichnend für das applikative Niveau. Aus Zeitgründen will ich auf den Plankalkül hier nicht im Detail eingehen, eine eingehende Würdigung aus moderner Sicht ist 1972 nach dem Erscheinen von Zuse's Autobiographie [1], geschehen ([3] , [4]).

Im Zusammenhang mit dem Plankalkül soll aber hier besprochen werden, wie der Übergang vom Formular zum Rechenplan, von der Formel zum sequentiellen Programm geschehen ist, wie der oben erwähnte "Superplan" funktionieren sollte, der automatisch dieses leistet. Mit der Postulierung einer solchen mechanischen Möglichkeit wäre Zuse der erste, der den Gedanken der "automatischen Programmierung" untersucht hat, wobei hier klarerweise "automatisch" nicht bedeutet, daß der Rechenplan aus dem Nichts entsteht, sondern aus der Formel. Wir kennen Rutishauser als den Vorreiter der "Automatischen Rechenplanfertigung". Es gibt aber bereits erste Anzeichen bei Zuse.

Da ist zunächst auffallend, daß eines der wenigen größeren Beispiele, die Zuse mit dem Plankalkül behandelt, die Entscheidung darüber ist, ob eine Formel ( des Aussagenkalküls) wohlgebildet ist (Abb.10). Diese Aufgabe (mit naheliegender Ausweitung auf beliebige Träger) ist als Vorform der Zerteilungsaufgabe (parsing), die den Kern der Formelübersetzung darstellt, bekannt. Des weiteren können wir aus Bemerkungen in Zuses Kommentar zum Plankalkül ([5], Kommentar, S. 5, S. 32-33) entnehmen, daß Zuse für die automatische Programmierung sowohl an ein "Programmfertigungsgerät" wie auch an eine algorithmische Beschreibung desselben dachte. Diese Beschreibung wäre aber einem Superplan auf einer (speicherprogrammierten!) algorithmischen Rechenmaschine ([6], S. 520), von der Zuse 1943/44 träumte, äquivalent gewesen und wohl im Plankalkül abgefaßt worden.

Zuse könnte also, wenn er sich dazu bekennt, auch als Vorläufer der Formelübersetzung angesehen werden, einer Aufgabe, mit der sich nach Rutishauser und Böhm ja auch Samelson und ich beschäftigten. Diesbezügliche Unterlagen Zuses waren uns allerdings 1955 nicht bekannt, auch hatte Rutishauser

V	① $R(V) \supseteq R$		
S	$\begin{matrix} o & o \\ m\sigma & o \end{matrix}$		
V	② $Az(V) \supseteq \& R$	③ $V \supseteq Z$	④ $o \supseteq \varepsilon$
K	$\begin{matrix} o & o \\ o & o \end{matrix}$	$\begin{matrix} o & o \\ o & o \end{matrix}$	$\begin{matrix} o & o \\ o & o \end{matrix}$
S	$\begin{matrix} \sigma & o \\ \sigma & \sigma \end{matrix}$	$\begin{matrix} \sigma & \sigma \\ \sigma & \sigma \end{matrix}$	$\begin{matrix} 1 & n \\ & \end{matrix}$
V	W $\left[ \begin{matrix} \textcircled{5} \mu x \left[ \begin{matrix} x \in V \& x \neq V \\ o & o \end{matrix} \right] \supseteq Z & \textcircled{6} Sq(Z, Z) \supseteq \& R \\ \sigma & m\sigma & \sigma & \sigma & \sigma & \sigma & o \end{matrix} \right]$		
K	$\begin{matrix} 1 & 1 \\ \sigma & \sigma \end{matrix}$		
S	$\begin{matrix} \textcircled{7} K1a(Z) \rightarrow (\varepsilon + 1 \supseteq \varepsilon) & \textcircled{8} K1z(Z) \rightarrow (\varepsilon - 1 \supseteq \varepsilon) \\ 1 & 1 \\ \sigma & \sigma \end{matrix}$		
V	$\begin{matrix} \textcircled{9} \varepsilon \geq o \supseteq \& R & \textcircled{10} Z \supseteq Z \\ o & 1 & o \\ o & \sigma & \sigma \end{matrix}$		
S	$\begin{matrix} \textcircled{11} Sz(Z) \supseteq \& R & \textcircled{12} \varepsilon = o \supseteq \& R \\ o & o & o \\ o & o & o \end{matrix}$		

Abb. 10

uns nie einschlägige Hinweise gegeben. Zuses Artikel von 1962 ([6], S. 520) enthält keine Angaben und erst die Autobiographie von 1970 [1] und die Veröffentlichung des Plankalküls 1972 durch die GMD ([5]) mit Zuses Kommentar brachten erste Andeutungen.

### Zuse als faustischer Geist

Zuse hat sich selbst (s. o.) als faustischen Geist bezeichnet. Diese Charakterisierung trifft wohl auf ihn wie auf selten jemand zu. Damit scheint es auch einherzugehen, daß trotz äußerlicher Ehrungen und Würdigungen sein erfinderisch-schöpferisches Gesamtbild auf unsere oberflächliche Generation, insbesondere auf die Jungen, nicht den notwendigen tiefen Eindruck hinterlassen hat. Zuse ist immer noch aktuell, im Plankalkül befinden sich präalgorithmische Elemente, die heute als nicht-deterministische Hilfsmittel zur Entscheidungsverzögerung beim Programm-entwurf ihre große Bedeutung erweisen. Wo wird an deutschen Hochschulen der Plankalkül, wenn schon nicht in einer Vorlesung, so doch wenigstens in einem Seminar behandelt? Die nachrückende Generation von Informatikern (nicht von Elektro-Ingenieuren!) darauf aufmerksam zu machen, daß sie sich mit Zuse beschäftigen sollen, dazu sollte dieser Vortrag anregen. Konrad Zuse aber, dem Altmeister, erbiere ich respektvoll meine Reverenz.

Literatur

- [1] K. Zuse: Der Computer, mein Lebenswerk. Verlag Moderne Industrie, München 1970.
- [2] Fr. A. Willers: Methoden der praktischen Analysis. De Gruyter, Berlin 1928.
- [3] F. L. Bauer, H. Wössner: Zuses "Plankalkül", ein Vorläufer der Programmiersprachen- gesehen vom Jahre 1972. Elektronische Rechenanlagen 14 : 3, 111-118 (1972).
- [4] F. L. Bauer, H. Wössner: The "Plankalkül" of Konrad Zuse: a Forerunner of Today's Programming Languages. Comm. ACM 15 : 7, 678-685 (1972).
- [5] K. Zuse: Der Plankalkül. BMBW-GMD-63, 1972.
- [6] K. Zuse: Entwicklungslinien einer Rechengenäte-Entwicklung von der Mechanik zur Elektronik. In: Hoffmann, W. (ed.): Digitale Informationswandler. Friedr. Vieweg und Sohn, Braunschweig 1962.

**Konrad Zuse - Sein Wirken allein und mit anderen**

Professor Dr.-Ing. Hermann Flessner

Die erste funktionsfähige programmgesteuerte Rechenanlage mit der eigens dazu ausgewählten Zahlendarstellung und Schaltalgebra sowie der Plankalkül als allgemeine algorithmische Programmiersprache sind von meinen Herren Vorrednern als herausragende Entwicklungen Konrad Zuses bezeichnet worden. Die Leistungen und die Persönlichkeit Konrad Zuses waren aber nur unvollständig gewürdigt, wenn nicht wenigstens einige weitere, vor allem anwendungsbezogene Arbeiten und seine besonders positive und angenehme Einstellung zu Freunden, Mitarbeitern, Partnern und Kunden beschrieben würde - denen gegenüber er nie ein unterscheidbares Verhalten erkennen ließ. Es mag vermessen klingen, wenn ein Jüngerer öffentlich eine derartige Aussage wagt. Sie wurde mir gegenüber aber belegt. Auch hatte ich den Vorzug, in den vergangenen zwanzig Jahren selbst zunächst Kunde und dann Partner von Konrad Zuse zu sein. Der Bauingenieur Zuse mit respektabler Erfahrung als Konstrukteur und Statiker hatte es natürlich nicht schwer, sich in die Vorstellungen und Wünsche von Versuchstechnikern, Vermessungs-Ingenieuren und Brückenbauern zu versetzen. Sicher ist damit auch, wenigstens zum Teil, die große Zustimmung zu erklären, die die elektronischen Rechenanlagen aus seiner Firma nach dem Kriege erfuhren, nachdem sie ab ungefähr 1954 nicht mehr vorwiegend an Universitäten ausgeliefert wurden.

Da war die Relais-Rechenanlage Z11, als erste in Serie für optische Industrie und für das Vermessungswesen, aber auch andere Einsatzgebiete entwickelt. Wenn diese inzwischen an einigen Stellen außer Betrieb genommen wurden, dann nicht etwa wegen nachlassender Betriebssicherheit. Wahre Oldtimer tun noch heute zuverlässig ihren Dienst!

Der Röhrenrechner Z22 wurde 1955 nach Plänen von Theodor Fromme aufgelegt. Diese Anlage hatte einen analytischen Befehlscode, der sie außerordentlich flexibel und leistungsfähig machte. Dabei hatte die erweiterte Version Z22 R nur 32 Kernspeicherzellen, die nicht einmal alle mit ganzer Wortlänge genutzt werden konnten, und nur 8.192 Trommelspeicherzellen, also 8K Worte zu 38 Bit. Trotzdem schaffte es Prof. Bauer - Sie haben ihn eben gehört - mit seinen Mitarbeitern der Universität Mainz einen guten ALGOL 60-Übersetzer für dieses Programm (*für diesen Rechner !*) zu schreiben, was auch an dem günstigen Konzept der Z22 lag.

Als Anwender und auch als Kunde programmierte man selbst, zunächst ausschließlich, später überwiegend im Maschinen-Code, dem "Freiburger Code", benannt nach der Universität, an der er entstand. Der Name Otto Suppes ist hiermit untrennbar verknüpft. Er war die Seele des Programmiergeschäftes der Firma Zuse und blieb bei den unmöglichsten Wünschen der Kunden stets gelassen. Überhaupt war das Verhältnis zwischen den Firmenangehörigen und Auftraggebern absolut unkompliziert; gleichgültig, ob man mit Herrn Dr. Zuse, dem Chef, oder einem Techniker sprach. Manchmal war es gar nicht einfach für einen Außenstehenden, zu erkennen, ob man es mit einem Zuse-Angestellten oder mit einem Besucher zu tun hatte, zum Beispiel auf der alljährlich stattfindenden Benutzertagung. Apropos Benutzergemeinschaft: ihr fühlten sich alle Zuse-Kunden zugehörig. Über sie wurden neue Rechnerkonzepte vorgestellt und Anwenderwünsche vorgebracht. Immer war Dr. Zuse selbst aktiv an der Betreuung der Tagungsteilnehmer beteiligt.

Wurde Ende der fünfziger, Anfang der sechziger Jahre bei der Firma Zuse eine Rechenanlage in Auftrag gegeben - inzwischen war die auf Transistor-Technik umgestellte, vom Konzept her aber der Z22 R vergleichbare neue Anlage Z23 entstanden - dann kümmerten sich in der Regel die für die Betreuung der Anlage auf Seiten der Kunden zuständigen Personen selbst um den Entstehungsablauf des eigenen Computers. Mit Dr. Zuse wurde diskutiert, mit seinen Mitarbeitern auf dem Prüfstand die Entstehung beobachtet und notwendige Wartungsarbeit mit Wartungstechnikern schon vorbesprochen, man machte sich bereits im Werk mit seiner späteren Rechenanlage vertraut. Vielfach wurde ein Wartungstechniker von der Firma Zuse in den Wirkungsbereich der neuen Anlage mit übernommen. Wenn dieses auch häufig nach Abwerbung schmeckte, so geschah es doch meines Wissens stets im Einvernehmen mit der Geschäftsleitung der Firma Zuse, jedenfalls beeinflusste diese Personal-Fluktuation die guten Beziehungen zur Firma nicht. Die leidigen Finanzierungsfragen haben der nicht sehr finanzstarken Firma Zuse stark zugesetzt! Große und mächtige Konkurrenten konnten ihre Anlagen vermieten, den Bau zu einem erheblichen Teil also selbst vorfinanzieren. Dieses konnte Zuse nur in seltenen Ausnahmefällen, die Banken als Kreditgeber spielten oft nicht mit.

Die Benutzergemeinschaft war auch gedacht als Forum für den Austausch von Anwenderprogrammen. Seit ungefähr 1960 gab es für die verschiedenen Einsatzgebiete Rechenprogramme, die bereits ihr Geld verdienten. Im Vermessungswesen und in der Bautechnik waren Berechnungsverfahren von je her schon formalisiert, die Berechnungen wurden erleichtert oder wenigstens vorbereitet durch Formulare und Matrizen-Schreibweise, alles Vorgehensweisen, die sich für das Programmieren bestens eignen. Diese Formularrechnung ist stumpfsinnig und kann nicht gerade als menschenfreundlich bezeichnet werden. Zuse hat dieses als auslösend für seinen Entschluß beschrieben, hier durch Konstruktion eines Rechengerätes Abhilfe zu schaffen. Er sagte, ich zitiere: "Es wollte mir nicht in den Kopf, daß lebendige, schöpferische Menschen ihr kostbares Leben mit derart nüchternen Rechnungen verschwenden sollten".

Auf dem vorbenannten Anwendungsgebiet konnte Zuse alle auftretenden Fachfragen selbst authentisch und erschöpfend beantworten, denn, bis er seine erste Rechenanlage Z1 fertigstellte, hatte er 1935 als Angestellter der Henschel-Flugzeugwerke ausgiebig Gelegenheit, sich mit statischen Berechnungen abzulassen. Kunden aus diesem Fachgebiet fanden daher in Konrad Zuse einen stets verständnisvollen Partner.

Die Erwartung, daß die Verbindung der Kunden untereinander zu einem Programmaustausch, evtl. bei Ausgleich der jeweils gegenseitig entstandenen Kosten, führen würde, erfüllte sich leider nicht. Desto erstaunlicher war die Bereitwilligkeit Zuses und seiner Mitarbeiter, für Kunden Anwendungsprogramme selbst zu erstellen oder auf Kosten der Firma Zuse erstellen zu lassen. Im Grunde genommen war diese Bereitschaft für das Haus Zuse ruinös.

Schließlich war allen klar, daß die dadurch hervorgerufenen großen Kosten nicht ohne weiteres auf die immerhin recht teuren elektronischen Rechenanlagen umgelegt werden konnten. Heftige Diskussionen für und wider das "Bundling" bzw. "Unbundling" entspannen sich in den Jahrestagungen der Benutzergemeinschaft, also um das Mitliefern von Programmen mit den Rechnern gegen Zahlung eines Gesamtpreises einerseits bzw. das getrennt Bereitstellen von Hardware und Software andererseits im Unbundling.

Zunehmend erkannten die Anwender, daß mit zu hoch gesteckten Forderungen bezüglich der Programmbereitstellung die Firma Zuse übermäßig finanziell belastet würde. Gern nahm man aber das fast selbstverständliche Angebot von Zuse in Anspruch, auf noch nicht ausgelieferten Rechnern in der Fabrik oder in den Niederlassungen kostenlos selbst seine Programme schon dann austesten zu dürfen, wenn die Anlage noch nicht im eigenen Hause stand. Nachdem hier in Hamburg in der Magdalenenstraße 1, in der Nähe der Außenalster in einer schönen alten Villa im November 1962 eine Niederlassung mit einer Anlage Z23 eröffnet worden war, konnte ich selbst

von diesem großzügigen Angebot als Hochschulangehöriger Gebrauch machen und meine ersten Programme für meine Dissertation nachts kostenlos laufen lassen. Die damaligen Leiter der Niederlassung Hamburg drückten einem abends den Hausschlüssel in die Hand, stellten eine Cola auf den Tisch und ließen einen mit dem Rechner allein - wo gab es das sonst, wo gibt es so etwas noch heute?

Es würde zu weit gehen, alle Einsatzgebiete aufzuführen, in denen von Beginn der Auslieferung der ersten Anlagen an und zum Teil auch noch heute, Zuse-Anlagen mit Erfolg und gewinnbringend arbeiten.

Doch sind fast überall die Erfolge nur möglich gewesen, weil die Auffassung Zuses von Partnerschaft, auf seine Mitarbeiter übertragen, bei seinen Kunden immer das Gefühl hinterließ, daß im entscheidenden Augenblick auch Unmögliches verwirklicht wird. Zwei Dinge sollten noch erwähnt werden:

die Entwicklungen Zuses auf dem Gebiet des automatischen Zeichnens und die Entwicklung eines Rechnertyps, den Zuse selbst für eine verpaßte Chance hält, nämlich die Dezimalrechenanlage Z31.

Die enge und langjährig fruchtbare Zusammenarbeit Zuses mit Herrn Dr. Lang, als Regierungsdirektor tätig in der Landesvermessung Hessen, hatte - wie Ingenieure sagen - zu einem "Pflichtenheft" für die Anforderungen an einen automatischen Zeichentisch geführt. Zuse entschied sich auch hier für ein digital gesteuertes Gerät. Er griff eine schon früher von ihm entwickelte Idee auf und konstruierte ein Binär-Stufengetriebe, bei dem die unterschiedlichen Geschwindigkeiten der Zeichenstift-Fahrbewegungen über ein durch Binär-Faktoren abgestuftes Getriebe gesteuert wurde. Die durch das Getriebe des Zeichentisches beeinflussten Geschwindigkeiten konnten darüber hinaus überlagert und kombiniert werden.

Zuses Entwurf galt für ein mechanisches Getriebe. Das erforderte besondere Überlegungen. Wird beispielsweise ein Rechteck mit der vom Antrieb her möglichen Höchstgeschwindigkeit gezeichnet, dann treten beim Richtungswechsel des Fahrstiftes in den Ecken zwangsläufig sehr hohe Verzögerungen oder Beschleunigungen auf. Sie erzeugen, im Verein mit der bewegten Masse, unerträglich große Kräfte. Man stelle sich vor, man führe mit dem Auto in eine scharfe Kurve, ohne die Fahrtgeschwindigkeit rechtzeitig zu mindern.

Zuse löste die Aufgabe mit einem von ihm modifizierten Malteserkreuz-Getriebe, das vor Richtungsänderungen des Fahrstiftes die Geschwindigkeit stetig reduzierte und danach genauso wieder steigerte. Noch heute arbeiten mehr als zehn Jahre alte Zuse-Graphomaten mit diesen mechanischen Getrieben einwandfrei. Inzwischen wurden aber Schrittmotoren verwendet - auch bei Zuse - und andere Verfahren eingesetzt, z.B. Magnetführungen, die erheblich größere Geschwindigkeiten zulassen. Das mechanische Getriebe hat inzwischen historischen Wert.

Die Dezimalrechenanlage Zuse Z31.

Sie war 1960 entstanden und als kleine, für kommerzielle Zwecke einsetzbare Maschine gedacht und sollte einmal in großer Stückzahl gebaut werden. Man war, nach Zuses Worten, aber zu sehr auf Maschinen nach dem Dual-Prinzip eingestellt. Die Ingenieure Zuses, aber auch die mit ihnen zusammenarbeitenden Kunden, waren nicht zu bremsen. Die Maschine wurde immer größer, bis ein großes leistungsfähiges und einwandfrei arbeitendes Gerät entstand, das den Maschinen der Klasse Z22, Z23 und der später entstandenen Z25 heftige Konkurrenz machte. Es ergab sich wieder einmal das Problem, die notwendigen Betriebsprogramme zu entwerfen und zu schreiben. Auch war es für Zuse untragbar, zwei völlig unabhängige ALGOL-Übersetzer zu erstellen.

Die Z31 erwies sich aber als eine wirkliche Universal-Maschine. Sowohl Ingenieure mit ihren

ausgesprochenen naturwissenschaftlichen-technischen Aufgaben - beispielsweise die Mitarbeiter der Hamburgischen Schiffbau-Versuchsanstalt - als aber auch Kunden, die die Z31 ausschließlich für kommerzielle Zwecke einsetzten, waren des Lobes voll. Es entstand die grotesk anmutende Situation, daß die Kunden die Anlage lobten, während die Angehörigen Zuses, insbesondere die Vertriebs-Ingenieure, ihre Nachteile herausstellten. Die schon 1962 an die Hamburgische Schiffbau-Versuchsanstalt ausgelieferte Z31 war übrigens die 200. Anlage der Firma Zuse überhaupt. Bis vor wenigen Wochen hat sie einwandfrei ihren Dienst verrichtet, wurde jetzt abgebaut und soll in einer Sammlung der Universität Hamburg betriebsbereit vorgeführt werden. Wie gesagt, war die Z31 eingesetzt für komplizierte technische Aufgaben, wie Strömungsberechnungen und Festigkeitsuntersuchungen. Andererseits war sie in einigen Exemplaren in den Datenverarbeitungs-Zentren der Kassenärztlichen Vereinigungen zu finden. Die erste wurde aufgestellt in Bad Segeberg in der Kassenärztlichen Vereinigung Schleswig-Holsteins. Auch hier hat sie anstandslos ihren Dienst getan und konnte selbst mit sehr viel moderneren Maschinen noch lange konkurrieren. In den Kassenärztlichen Vereinigungen wird das Abrechnungswesen der Ärzte und Krankenkassen koordiniert und durchgeführt. Umfangreiche Erfassungs- und Datenhaltungsprobleme treten dort auf. Für die Erfassung von Daten wurde bei Zuse, wiederum in enger Kooperation zwischen Anwendern und der Firma, das Datenerfassungsgerät Z16 entwickelt. Noch heute sind ca. 150 Stück Z16-Erfassungsgeräte über die ganze Bundesrepublik verteilt im Einsatz und leisten anstandslos ihren Dienst. Aus der Vielfalt der Einsatzgebiete von Zuse-Geräten seien nur einige herausgegriffen, teilweise recht gegensätzliche:

- Vermessungswesen und halbautomatische Tachymetrie in diesem Gebiet,
- Bekleidungsindustrie,
- das schon genannte Bauwesen in fast allen Bereichen,
- die Steuerung von Walzenstraßen - schon Mitte der 60er Jahre,
- Schiffbau und Flugzeugbau,
- Abrechnungswesen, schon genannt die Arbeiten in den Kassenärztlichen Vereinigungen, aber auch im Groß- und Einzelhandel,
- Versuchswesen, beispielsweise in der Bundesanstalt für Materialprüfung in Berlin (BAM),
- Einsatz in Kraftwerken und zur Steuerung ganzer Kraftwerksblöcke bei den Energieversorgungs-Unternehmen.
- Aber auch in der Biometrie, weiten Bereichen der Naturwissenschaften und sogar in der Linguistik kamen Anlagen von Zuse schon in den 60er Jahren zum Einsatz.
- Auch die Streckenführung und Querschnittsgestaltung der Olympia-Bobbahn für die Olympischen Spiele 1964 in Innsbruck wurde nach fahrdynamischen Grundsätzen mit einer Zuse-Anlage berechnet.

Dies alles geschah in einer Ära, die, wie es schien, 1971 ein Ende haben sollte. In der Zeit bis dahin waren aber Betriebsangehörige der Firma - Konrad Zuse eingeschlossen - und die Kunden Partner, ja sogar Freunde geworden - über die Auflösung der Zuse-KG und Übernahme durch Siemens hinaus.

Die grundlegenden Entwicklungen bis zur Anwendungsreife stammen aus Zuses eigenen Ideen und seiner eigenen Tatkraft. Die Betonung seiner Fähigkeit, mit allen gut zusammen arbeiten zu können, darf nicht darüber hinwegtäuschen, daß er es sich leistete und leisten konnte, andere neben sich erfolgreich sein zu lassen. Auch in schwierigen Zeiten versuchte er, seinen Humor zu behalten und auf seine Freunde und Mitarbeiter sowie Kunden, zu übertragen.

Als seine Zeit als Unternehmer ein Ende fand, erschien sein Humor Außenstehenden manchmal ein wenig makaber. Wenn Konrad Zuse in gewissen Abschnitten seines Lebens enttäuscht war, hat er nie die Schuld bei anderen, sondern zunächst bei sich selbst gesucht. Enttäuschung schien ihn auch zu belasten, wenn seine Überlegungen einfach nicht verstanden wurden. Doch wie viele

seiner Gedankengänge waren und sind auch heute noch einfach der Zeit weit voraus, so daß eine Realisierung aus wirtschaftlichen Gründen, aber auch weil die Einstellung vieler Menschen auf diese Entwicklungen noch nicht genügend eingestimmt ist, nicht sofort erwartet werden kann.

Alle, die Konrad Zuse länger kennen, fasziniert seine Universalität. Er ist kein Spezialist, der sich nur um seine Erfindungen und deren technische Realisation kümmert. Er ist vielmehr allen schönen, insbesondere auch den musischen Dingen, zugetan, und wie ja wohl inzwischen auch bekannt geworden ist, ein fleißiger und beachteter Maler. Seine Bilder, heute mit seinem richtigen Namen signiert, trugen früher das Pseudonym "Kuno See". In seiner Studienzeit nannten ihn seine Freunde, vor allem die Kommilitonen in der Akademischen Vereinigung MOTIV "Kuno von und zu See", woraus er dann sein Signum bzw. Pseudonym ableitete: Kuno See. Als in Bad Hersfeld das neue Geschäftsgebäude fertiggestellt war und Professor Zuse oben auf dem Dach in einer Art Vogelbauer residierte, hatte er immer noch für seine Kunden und Freunde Zeit, sich mit Ihnen über ihre Sorgen und Pläne zu unterhalten.

Daß Konrad Zuse, alleine und mit anderen, Hervorragendes geleistet hat, habe ich versucht zu beschreiben, so gut es mir gegeben ist. Er ist ein hervorragender Solist, der viele Instrumente beherrscht, aber vorzüglich auch mit dem Orchester zusammenspielen kann - vielleicht ab und zu ein wenig zu moderne Musik, zu modern für den landläufigen Geschmack. So modern, daß sie leider nicht von allen schnell genug verstanden wird, oder sagen wir ruhig, von vielen nicht gemocht wird!

Mit mir erfüllt es meine Kollegen, unsere Studenten und viele Ihrer Freunde, von denen etliche heute hier versammelt sind, mit großer Freude und Genugtuung, daß wir Sie als den ersten Ehrendoktor unseres Fachbereiches in der Universität Hamburg auszeichnen dürfen, Sie, lieber Herr Zuse, dem im Laufe der letzten Jahre viele Ehren zu einer Zeit zuteil wurden, die nicht das Ende Ihres Schaffens einläuteten, sondern in der Sie selbst diese Ehrungen noch als Aufforderung verstehen, weiter tätig zu sein. In Ihrem schon genannten Buch sagten Sie, ich zitiere wieder, "es liegt in der Natur solcher Ehrungen, daß sie den Leistungen der Vergangenheit gelten. Ich würde es als die größte Ehrung sehen, wenn man auch meinen heutigen Ideen Aufmerksamkeit zuwenden - und mir bei ihrer Durchsetzung helfen würde."

Es wurden Ihnen schon viele Denkmäler errichtet. Daß Sie selbst noch kein Denkmal sind, sondern uns alle noch ständig zum Denken anregen können, sollten wir als Aufforderung auffassen, Sie auch weiterhin als Partner in Anspruch zu nehmen. Ich bin sicher, lieber Herr Zuse, daß mit mir alle Ihre Freunde hoffen, dieses noch sehr lange tun zu dürfen !

UNIVERSITÄT HAMBURG

Der Fachbereich Informatik der Universität Hamburg verleiht

HERRN

PROF. DR.-ING. E.H.  
**KONRAD ZUSE**

in Anerkennung seiner bahnbrechenden Verdienste um die Informatik durch die Entwicklung der ersten arbeitsfähigen programmgesteuerten Rechenanlage und durch sein wegweisendes Konzept einer algorithmischen Sprache

GRAD UND WÜRDE EINES  
**DOKTORS  
DER NATURWISSENSCHAFTEN  
EHRENHALBER**

Zum Zeugnis dessen wird diese mit dem Siegel der Universität versehene und vom Sprecher des Fachbereichs unterzeichnete Urkunde ausgestellt.

Hamburg, den 10. Oktober 1979



*W. G. B.*

Der Sprecher des Fachbereichs  
Informatik

## Die Emanzipation der Datenverarbeitung

Konrad ZUSE

Die Entwicklung des Computers wurde zunächst durch Ingenieure und Wissenschaftler vorangetrieben. Babbage, der eigentliche Vater des Computers, war Mathematiker. Auch die späteren Entwicklungen in den USA, die etwa z.Zt. des ersten Weltkrieges begannen, wurden von Wissenschaftlern und Ingenieuren wie Stibitz, Aiken, Eckert, Mauchly, durchgeführt mit der Zielsetzung, leistungsfähige Instrumente für ihre eigenen Probleme zu schaffen. Ich selbst war Bauingenieur und hatte von vornherein ingenieurmässige Aufgaben im Auge. Damit war eine enge Verbindung zwischen Rechenmaschinenteknik und Mathematik gegeben. Sie erstreckte sich nicht nur auf die mit diesen Geräten zu lösenden Aufgaben, sondern auch auf den Entwurf und die theoretischen Grundlagen der Konstruktion der Geräte selbst.

Es sind dabei verschiedene Auffassungen möglich. Rückblickend auf meine eigenen Entwicklungen kann ich mit Befriedigung feststellen, dass z.B. die Anwendung der mathematischen Logik auf den Schaltungsentwurf sehr fruchtbar war. Das regte dazu an, eine Theorie des allgemeinen Rechnens zu schaffen, die einerseits auf exakte mathematische Disziplinen beruht, andererseits jedoch voll auf praktische Aufgaben eingestellt war. Es bestand bereits in den vierziger Jahren eine harmonische Zusammenarbeit mit Vertretern der mathematischen Logik. Mein mathematischer Mitarbeiter, Herr Lohmeyer, kam aus der Schule des bekannten Logikers Heinrich Scholz in Münster, der - trotz der damaligen sehr abstrakten Einstellung - die neuen praktischen Anwendungsmöglichkeiten klar erkannte und positiv beurteilte.

Bezeichnend für unsere Entwicklung in Deutschland war, dass sie vollkommen Praxis-bezogen war. Es bestand ferner keine Verbindung zu den etwa gleichzeitig laufenden Arbeiten in den USA und in England. Als 1945 die Fortsetzung der Geräteentwicklung für uns zunächst unmöglich war, konzentrierte ich mich auf das Problem der Formulierung allgemeiner Rechenvorschriften. Das führte zum Plankalkül, den wir heute als erste algorithmische Sprache bezeichnen können.

Es lässt sich heute überblicken, welche Verbindungen zwischen Rechenmaschinenentwicklung und mathematischen Disziplinen ausser den erwähnten eigenen Arbeiten bestand.

**Turing** hatte seine Arbeit über die Berechenbarkeit von Funktionen veröffentlicht.

**Shannon** hatte als einer der ersten (abgesehen von einer Arbeit von Hansi Piesch) eine Abhandlung über Schaltalgebra veröffentlicht.

**Stibitz** hatte einen Relaisrechner für komplexe Zahlenrechnungen entwickelt, der weitgehend Ähnlichkeit mit den Geräten Z3 und Z4 aufwies.

**Aiken** hatte mehrere Geräte gebaut und dabei auch Ideen in Richtung Schaltalgebra entwickelt.

**John v. Neumann** beeinflusste die Entwicklung, insbesondere in den ersten Nachkriegsjahren u.a. in Richtung Programmspeicherung.

**Norbert Wiener** formulierte unter dem Namen Kybernetik allgemeine Ideen zu Steuerungsproblemen.

Von verschiedenen Wissenschaftlern wurden ferner die **Informationstheorie** und die **Automatentheorie** entwickelt.

Wir haben heute genügend Abstand, um diese Arbeiten analysieren zu können. Die Ideen von Turing haben inzwischen Berühmtheit erlangt. Die "Turing-Berechenbarkeit" ist bei Theoretikern zu einem festen Begriff geworden. Allerdings sind seine Arbeiten sehr abstrakt und wenig Praxis-bezogen und haben daher die Computerentwicklung kaum beeinflusst.

Der Gedanke der Schaltalgebra hat sich heute allgemein durchgesetzt. Allerdings bezog sich Shannon in seiner ersten Veröffentlichung noch nicht unmittelbar auf den Aussagenkalkül. Heute ist es jedoch selbstverständlich, von "Und-Gliedern", "Oder-Gliedern" usw. zu sprechen.

Wie weit Stibitz die mathematische Logik als Hilfsmittel benutzte, ist mir nicht bekannt.

Dagegen hat John v. Neumann die Computerentwicklung stark beeinflusst. Die Idee, auch das Programm als Objekt der Datenverarbeitung zu betrachten, ist allerdings im Plankalkül bereits voll vorweggenommen. J.v.Neumann kommt jedoch das Verdienst zu, den ersten Computer mit Programmspeicherung und Adressenumrechnung gebaut zu haben. Daher spricht man heute gern von der "J.v.Neumann-Maschine".

Die Ideen Norbert Wieners haben seinerzeit erhebliches Aufsehen erregt. Jedoch zeigt sich heute, dass er die Computerentwicklung selbst kaum unmittelbar beeinflusst hat. Eine Zeit lang bestanden zwar Bestrebungen, die Wissenschaft vom Computer als Teilgebiet der Kybernetik zu verstehen, eine Auffassung, die sich jedoch nicht durchsetzen konnte.

Die Informationstheorie wurde bereits vor der Entwicklung der Computer als Hilfsmittel der Nachrichtentechnik geschaffen. Das Hauptziel besteht darin, ein neutrales Maß für die in einer Nachricht enthaltenen Informationen zu gewinnen, z.B. um die Leistungsfähigkeit verschiedener Nachrichtenkanäle vergleichen zu können. Dabei spielen die Wahrscheinlichkeitsrechnung, die Störungsrechnung usw. eine wesentliche Rolle. Die Informationstheorie ist jedoch nicht die Theorie der Informationsverarbeitung. Bei ihr wird der Begriff "Information" quantitativ gedeutet und in der Informationsverarbeitung qualitativ. Die mangelhafte Unterscheidung dieser Auffassungen hat leider schon viel Verwirrung gestiftet.

Die Automatentheorie gehört heute mit zu den Werkzeugen der Computerwissenschaft. Sie ist allerdings ebenfalls sehr abstrakt. Sie kann das Verständnis mancher theoretischer Aspekte erleichtern. Die praktische Computerentwicklung ist durch sie jedoch kaum oder nur indirekt beeinflusst worden.

Die enge Verbindung des Computers mit den mathematischen Wissenschaften einmal von der Anwenderseite und zum anderen von der Entwicklungsseite her führte im Laufe der Jahre dazu, dass die Mathematiker den gesamten Fragenkomplex als ihre Domäne betrachten. Das hat manche Vorteile, birgt aber auch erhebliche Gefahren.

Für das gesamte mit der Entwicklung des Computers zusammenhängende Gebiet hat sich heute im englischen Sprachgebrauch das Wort Computer-science und bei uns das Wort Informatik durchgesetzt, welches ich zuerst - schon recht früh - aus dem Munde von Herrn Steinbuch vernommen habe.

Das Wort "Informatik" geht wohl auf die Begriffe Information und Automatik zurück und kennzeichnet daher gut die automatische Informationsverarbeitung. Information wird dabei allerdings, wie bereits erwähnt, im qualitativen Sinn aufgefasst. Die Vertreter der Informatik neigen stark zu theoretischen Auffassungen und Formulierungen. Die sehr sauber definierten Begriffe der Informationstheorie sind für einen Mathematiker ein Anreiz, sie zu übernehmen und weiter auszubauen.

Dies hat im Laufe der Jahre dazu geführt, dass die Informatik bzw. Computer-science zu einer sehr abstrakten Wissenschaft geworden ist, wobei wir leider feststellen müssen, dass Theorie und Praxis durchaus nicht immer gut harmonieren. Die Informatik konnte zwar eine erhebliche Selbständigkeit erlangen und sich so von den anderen Zweigen der Wissenschaft, insbesondere der Mathematik, weitgehend emanzipieren; jedoch ist es ihr nicht gelungen, den gesamten Komplex der mit der Computerentwicklung zusammenhängenden Fragen in der vollen Spannweite über reine Theorie, Programmierung, Software-Herstellung, Geräteentwicklung und Anwendung in befriedigender Weise zusammenzufassen. Oft sprechen Theoretiker und Praktiker eine verschiedene Sprache und haben grosse Schwierigkeiten, sich zu verstehen.

Nach meiner Auffassung ist dieses Auseinanderklaffen zwischen Theorie und Praxis jedoch weder von der Logik noch von der Anwendung her zu rechtfertigen. Es kann einfach nicht wahr sein, dass eine exakte Ausdrucksweise nur durch abstrakte theoretische Formulierungen möglich ist, und dass demgegenüber handliche praktische Darlegungen durch eine Unbestimmtheit in logischer Hinsicht erkauft werden müssen.

Informationsverarbeitung und Datenverarbeitung scheinen auf den ersten Blick hin dasselbe zu meinen. Tatsächlich klingt das Wort Information jedoch wissenschaftlicher; jedoch haben wir bereits gesehen, dass ein wesentlicher Unterschied zwischen der quantitativen Auffassung der Informationstheorie und der qualitativen Auffassung der Informatik besteht. Demgegenüber ist das Wort "Datenverarbeitung" weniger erblich belastet. Man spricht vom Informationsgehalt, aber nicht vom Datengehalt. Mathematiker werden sagen, dass im Gegensatz zum Begriff Information der Begriff Daten nicht klar definiert ist. Die strengen Definitionen der Information gelten aber nur im Rahmen der Informationstheorie und ihrer Zielsetzungen. Es mag für einen Mathematiker mitunter sinnvoll sein, bei den verschiedenen Datenverarbeitungsprozessen z.B. die Frage nach der Erhaltung der Information zu stellen. Man darf solche Gesichtspunkte aber nicht zu hoch bewerten; denn von einer Erhaltung der Information kann man allenfalls bei einer Daten- bzw. Nachrichtenübertragung sprechen, sofern man die Wirkungen von Störungen vernachlässigt. Bei den meisten Verarbeitungsprozessen wird jedoch im qualitativen Sinne aus der alten Information neue gewonnen. Bei dem Prozess geht im allgemeinen Information verloren; denn sehr oft werden aus einer Fülle von Daten nur wenige Daten als Ergebnis abgeleitet. Umgekehrt kann durch Zufallszahlen in einem Datenverarbeitungsprozess jederzeit beliebig viel Information neu geschaffen werden. Es ist dabei nicht von Belang, ob es sich dabei um echte oder um sogenannte Pseudo-Zufallszahlen handelt, sofern die Bildung der letzteren in ausreichender Weise unabhängig von dem eigentlichen Datenverarbeitungsprozess erfolgt. Der Informationsbegriff der Informationstheorie ist eben nur unter Vorbehalt in der Datenverarbeitung sinnvoll verwendbar. Das gilt auch für den damit eng zusammenhängenden Begriff "Informationsfluss".

Ich halte daher folgenden Weg für angemessen:

Der Begriff Daten bzw. Datum wird als Grundelement der Datenverarbeitungstheorie eingeführt. Das Basiselement ist bekanntlich das Bit. Es wird bewusst darauf verzichtet, das Bit etwa im Sinne der Informationstheorie als besonderen Fall für eine Entscheidung zwischen zwei Möglichkeiten abzuleiten. Im Gegenteil, das Bit als elementare Datenstruktur, die zwei Werte zulässt, führt zu der Konsequenz, dass damit eine zweiwertige Entscheidung formuliert werden kann. Alle weiteren Datenstrukturen werden auf dem Bit aufgebaut bzw. daraus zusammengesetzt.

Damit kommen wir nun auch zu einem elementaren Satz der Datenverarbeitung:

"Die Datenverarbeitung beginnt mit dem Bit".

Daten sind Objekte der Datenverarbeitung und jedes Datum hat eine Struktur, die im allgemeinen durch Definition gegeben ist. Sie müssen so aufgebaut sein, dass sie als ganze oder in ihren

Komponenten klar erfasst werden können. Da das Bit die elementare Datenstruktur ist, sind die Operationen mit Bits die elementaren Rechenoperationen. Wir wissen heute längst, dass es sich dabei um die Grundoperationen des Aussagenkalküls handelt. Damit ist der Weg für eine von der Basis her systematisch aufgebaute Lehre der Datenverarbeitung vorgezeichnet.

Diese Gedanken haben mich seinerzeit veranlasst, den Plankalkül zu entwickeln. Es ist die einzige konsequent nach diesen Gesichtspunkten aufgebaute algorithmische Sprache. Den Ausdruck "Kalkül" wählte ich seinerzeit, da meine Pläne ursprünglich noch wesentlich über eine algorithmische Sprache hinaus gingen, nämlich in Richtung einer geschlossenen unabhängigen Theorie der Datenverarbeitung. Ich kann an dieser Stelle nur durch einige Beispiele andeuten, was mir dabei wesentlich erscheint.

Daten haben stets eine Struktur. Diese lässt verschiedene "Werte" als konkret gegebenen Fall zu. Im Rahmen von Programmen können sie in allgemeiner Form unter einem Namen aufgeführt werden. Meistens ist die Zahl der möglichen Werte für eine Struktur endlich. Es lassen sich jedoch im Prinzip auch Strukturen mit unendlicher Zahl von Werten definieren. Eine solche Struktur, durch die z.B. natürliche Zahlen als Binärzahlen dargestellt werden können, ist im konkreten Fall stets ebenfalls endlich; jedoch besteht keine Grenze, bis zu welcher solche Werte dargestellt werden können. Der in der Mathematik eingeführte Begriff "potentiell unendlich" gilt auch hier.

In der modernen Mathematik konnte sich die Mengenlehre mit grossem Erfolg als wirkungsvolles Werkzeug durchsetzen. Man versteht dabei unter Menge eine Ansammlung wohl unterscheidbarer Elemente. Eine solche Menge kann auf verschiedene Weise beschrieben bzw. definiert werden, z.B. indirekt durch ein Prädikat, welches auf alle Elemente der Menge zutrifft. Die direkte Beschreibung besteht in der Nennung bzw. Aufzählung ihrer Elemente.

Vom Standpunkte der Datenverarbeitung ist dazu folgendes zu sagen: Mengen müssen durch Daten gekennzeichnet werden. Das kann indirekt durch eine mathematische Formel z.B. ein Prädikat in Sinne des Prädikatenkalküls der mathematischen Logik erfolgen. Auch diese Formel ist ein Datum. Am leichtesten verständlich ist die direkte Darstellung einer Menge durch eine Liste, in der alle zu der Menge gehörenden Elemente durch ihre Bezeichnung aufgeführt sind. Wir benötigen dafür die Definition der Datenstruktur für das einzelne Element und diejenige für eine Liste, als Aneinanderreihung von Elementen bzw. Gliedern. Dabei fällt uns sofort eine wesentliche Eigenschaft des datenmässigen Denkens auf. Die Datenverarbeitung kennt nur geordnete Listen, d.h. die Glieder einer Liste sind in einer bestimmten Reihenfolge aufgeführt. Man kann durch eine solche Liste eine Menge kennzeichnen. Der mathematische Begriff der Menge kennt jedoch als solcher keine Ordnung der Elemente. D.h., dass dieselbe Menge durch verschiedene Listen beschrieben werden kann. Alle diese Listen haben dieselbe Struktur; jedoch stellt jede der verschiedenen möglichen Anordnungen der Glieder in der Liste ein anderes Datum dar. Man kann nun auf der Struktur "Liste" eine Operation (Programm) definieren, mit dem Ergebnis "die Listen A und B enthalten dieselben Elemente". Die Listen sind also "mengenmässig äquivalent". Auf diese Weise lassen sich die Operationen der Mengenlehre durch Operationen der Listenverarbeitung darstellen. Man kann einen dem Mengenkalkül entsprechenden in sich geschlossenen Listenkalkül aufstellen. Die wichtigsten Ansätze dazu habe ich im Plankalkül bereits ausgearbeitet.

Ogleich eine Menge laut Definition an sich keine bevorzugte Ordnung ihrer Elemente kennt, wird von den Mathematikern doch oft eine Ordnungsrelation in die Definition einer Menge einbezogen. Das gilt z.B. für die Menge der natürlichen Zahlen. Sie wird erst durch die Ordnungsrelation zu einem sinnvollen Begriff. Es lässt sich wohl die Auffassung vertreten, dass die Menge der natürlichen Zahlen zwischen dem Begriff "Menge" und "Gruppe" steht. Der Umstand, dass

man auf Mengen Operationen mit bestimmten Eigenschaften definieren kann, hat zur Gruppentheorie geführt, die sich in ihren verschiedenen Modulationen als sehr fruchtbar erwiesen hat.

Datentechnisch können wir nun zunächst feststellen, dass z.B. zur Darstellung der Menge der natürlichen Zahlen eine der vielen möglichen Anordnungen der Glieder dadurch ausgezeichnet ist, dass sie der gegebenen Ordnungsbeziehung entspricht. Damit ist in ökonomischer Weise die Menge und die Ordnungsbeziehung festgelegt. Allerdings kann durch einen bestimmten Wert einer Datenstruktur, in diesem Falle eine Liste, stets nur eine endliche Menge aus der an sich unendlichen Menge der natürlichen Zahlen herausgegriffen werden.

Ebenso wie bei Mengen kann die Darstellung unendlicher Listen rekursiv bzw. durch ein generatives Programm zur Bildung des Nachfolgers erfolgen. Datentechnisch kommt allerdings hinzu, dass es im Prinzip beliebig viele Möglichkeiten der Darstellung von Zahlen gibt. Die einfachste Form stellen die Binärzahlen dar. Mit ihnen als Objekte bzw. Variable lassen sich alle Operationen mit Zahlen als Programme definieren. Der Übergang von endlichen zu unendlichen Zahlenmengen (sowohl für natürliche als auch für rationale Zahlen) kann wiederum durch rekursive Definitionen bzw. Programme erfolgen.

Der Mathematiker kann einwenden, dass die Axiome der Zahlenrechnung im Prinzip nur durch unendliche Zahlenmengen erfüllt werden können, was für eine Rechnung auf eine unendliche Genauigkeit hinausläufe. Der praktische Bau eines arithmetischen Rechenwerks ist, so gesehen, stets ein Kompromiss. Datentechnisch gesehen sind die Axiomensysteme der Mathematik oft nur eine Illusion. Sie können nur durch einen lediglich gedanklich ausführbaren Übergang zum Unendlichen erfüllt werden.

Baut man das Gebäude der Zahlenrechnung von vornherein nach datentechnischen Gesichtspunkten auf, so kann diesem Umstand durch exakte z.B. rekursive Definition Rechnung getragen werden. In diesem Sinn kann ein Kalkül der Datenverarbeitung vielleicht mehr Aussagekraft besitzen, als ein solcher der Informationsverarbeitung in Sinne der Mengenlehre.

Unter Datenstruktur versteht man wohl auch in der Informatik nicht immer dasselbe. Die Auffassung des Plankalküls konnte sich leider nicht durchsetzen. Dort bestehen Datenstrukturen aus klar gegliederten Baumstrukturen, die den Zugriff zu jedem Zweig durch ein Programm erlauben; alles, was darüber hinausgeht, erfordert ein auf solchen Strukturen definiertes Programmpaket. Es handelt sich dabei um eine ähnliche Erweiterung der Begriffe wie beim Übergang von der Menge zur Gruppe. Als Beispiel seien Paarlisten angeführt, die als bevorzugte Fälle der Auflistung einmal in bezug auf die Vorderglieder und zum anderen in bezug auf die Hinterglieder geordnet werden können. Mengenmässig sind beide Listen äquivalent, sie stellen jedoch verschiedene Daten dar. Eine in dieser Hinsicht neutrale Darstellungsform wäre eine Matrix, wie sie in bekannter Weise zur Darstellung von Relationen angesetzt werden kann. Sie ist jedoch bei umfangreichen Paarlisten zu unhandlich. Es lassen sich für solche Fälle generelle Programme für die Bildung einer "invertierten" Kartei bzw. Paarliste ansetzen. Sie bilden nur einen Teil des auf Paarlisten definierbaren generellen Programmpakets.

Es ist daher meines Erachtens nach besser, bei dem im Plankalkül verwendeten Begriff der Datenstruktur zu bleiben und darüber hinaus von Programmpaketen zu sprechen, die auf bestimmten Datenstrukturen definiert sind. So gesehen könnte der Begriff Datenverarbeitung etwas anderes bedeuten als der Begriff Informatik; jedoch liegt kein Grund vor, beide Begriffe streng zu unterscheiden, da ja "Informatik" wohl noch nicht streng definiert ist. Wesentlich ist dabei lediglich, dass sich die Datenverarbeitung oder eine entsprechend aufgefasste Informatik von dem manchmal überstarken Einfluss der theoretischen Mathematik emanzipiert. Wie ich schon betonte, braucht sie dadurch an logischer Strenge nichts zu verlieren; im Gegenteil können

meines Erachtens manche Aspekte der praktischen Datenverarbeitung klarer dargestellt werden.

Nun möchte ich noch auf einige Perspektiven eingehen, die eine solche Emanzipation der Datenverarbeitung von der Anwendung her rechtfertigen.

Wenn auch die ersten Computer von Ingenieuren und Wissenschaftlern für ihre Aufgaben geschaffen wurden, so können wir doch heute feststellen, dass 80 bis 90 % aller Computer für kommerzielle und Verwaltungsaufgaben eingesetzt werden. Solange die Wissenschaftler mit ihren Computern unter sich blieben, wurden ihre Arbeiten von der übrigen Welt noch nicht sehr ernst genommen. Aufgaben, die mit den Computern gelöst werden konnten, waren genügend da. Niemand wurde arbeitslos. Im Gegenteil, der Beruf des Mathematikers nahm einen ungeahnten Aufschwung. Nie zuvor wurden von ihnen so viele gebraucht.

Erst das Vordringen in Bereiche, die letzten Endes das gesamte wirtschaftliche und gesellschaftliche Leben umfassen, hat die Menschheit aufmerksam werden lassen. Das wird durch die Einführung der integrierten Schaltkreise und Minicomputer noch verstärkt. Heute kann niemand mehr an der Entwicklung vorbeigehen; aber all diese Arbeiten, die da von den Computern übernommen werden, sind Datenverarbeitungsprozesse. Gerade dabei brauchen wir eine Sprache, die alle, die damit zu tun haben, verstehen. Letzten Endes entwickeln wir uns in Richtung auf die Verwaltung und Beherrschung des gesamten Erdballs mit Hilfe gigantischer Datenverarbeitungssysteme. Da ist noch enorm viel zu tun. Regieren ist die schwerste aller Arbeiten. Die Möglichkeiten, hierfür Computer nicht nur rein verwaltungsmässig, sondern auch im Sinne der künstlichen Intelligenz einzusetzen, werden noch kaum erkannt. Wir dürfen uns durch Fehlschläge auf diesem Gebiet nicht entmutigen lassen. Die integrierte Schaltkreistechnik bietet hier neue Möglichkeiten der assoziativen Datenverarbeitung, die wir uns in den nächsten Jahren erst erarbeiten müssen.

Die theoretische Physik stellt heute eines der schwierigsten und abstraktesten Gebiete der Wissenschaft dar. Unter gewaltigem formalen Aufwand werden Mittel der Mathematik eingesetzt, um im Sinne von Faust zu erkennen, was die Welt im Innersten zusammenhält. Es hat mich sehr gefreut, dass im Jahre 1978 in Berlin eine gemeinsame Jahrestagung der Physiker und Informatiker stattfand. Die enge Verbindung zwischen beiden Gebieten wurde dabei aber erst instinktiv geahnt. Die Computer sind für die Physiker heute bereits ein sehr wirkungsvolles Hilfsmittel; jedoch denkt man in den Kreisen der theoretischen Physiker noch nicht daran, im Kosmos einen gewaltigen Datenverarbeitungsprozess zu sehen. Die Voraussetzungen für eine solche Betrachtungsweise müssen auch erst geschaffen werden. Meine eigenen Arbeiten auf dem Gebiete des "Rechnenden Raumes" liegen in dieser Richtung. Ich hatte Gelegenheit, an der Universität Braunschweig bei Herrn Professor Vollmar einige Modelle von zellularen Automaten mit dem Computer zu simulieren, die in dieser Richtung liegen. Die Ergebnisse sind noch keineswegs sensationell sondern haben zunächst mehr spielerischen Charakter. Jedoch hoffen wir, dass sich aus diesen Spielereien eines Tages neue Erkenntnisse ergeben.

Es sei mir erlaubt, einige kühne Perspektiven in dieser Richtung zu skizzieren. Im Sinne eines digitalisierten Kosmos ist die Quanten-Theorie nur ein Schritt in Richtung der Überwindung des Denkens in kontinuierlichen Räumen, Differentialgleichungen usw. Die Physiker müssen sich streng an das durch Versuche erwiesene Gebäude der Erscheinungen und Theorien halten. Bei der Idee des Urknalls setzen sie also auch voraus, dass damals unsere heutige Physik mit all ihren physikalischen Grössen, Elementarteilchen usw. bereits fertig vorlag. Unsere heutigen Kenntnisse erlauben zunächst keinen anderen Schluss. Aber vielleicht wird man diesen Standpunkt in einiger Zeit für ebenso naiv halten wie das Bild der Naturwissenschaften von der Entstehung der Arten vor Darwin. Vielleicht müssen wir für die Entwicklung des Kosmos eine ähnliche Evolutionstheorie ansetzen, wie in der Biologie. Dem Urknall im Sinne einer bei einem

bestimmten Punkt der Entwicklung einsetzenden gewaltigen Ausdehnung ging vielleicht ein komplizierter Datenverarbeitungsprozess auf beschränktem Raum voraus. Die Natur probierte Vieles aus. Das Elektron blieb übrig. Das alles ist zunächst nur Phantasie; aber es ist zu hoffen, dass neue Aspekte wie die der Datenverarbeitung, eines Tages zu grundsätzlich neuen Theorien führen können.

Auf dem Gebiete der Automation wird die Datenverarbeitung bereits mit Erfolg eingesetzt. Allerdings stehen wir auch hier am Anfang. Wir sind noch weit davon entfernt, den Aufbau ganzer Organismen aus einer Keimzelle heraus mit einem Programm steuern zu können. Die Natur ist uns mit der in der DNA-Kette gespeicherten Information weit voraus. All das sind letzten Endes auch Datenverarbeitungsprozesse.

Bei der Evolution der biologischen Organismen spielt die Selbstreproduktion eine entscheidende Rolle. Auch in dieser Richtung steht unsere Technik und Wissenschaft erst am Anfang. Unsere Produktionsmethoden unterscheiden sich grundsätzlich von denen der Natur. Trotzdem bin ich davon überzeugt, dass wir noch vor revolutionären Wandlungen stehen. Die sich selbst reproduzierenden Systeme wurden von mir in bezug auf die konstruktiven Aspekte untersucht. Als fernes Ziel schwebte mir dabei etwa vor, eine Werkstatt zu schaffen, die ihre eigenen Werkzeugmaschinen nachbauen kann. Das Ziel kann verschieden weit gespannt werden. Ein unmittelbar erreichbares Nahziel waren die eigenbezogenen Systeme, bei denen nicht auf volle Selbstreproduktion Wert gelegt wird; sondern lediglich wie in einer autarken Wirtschaft, ein Satz von Werkzeugen oder Produktionsmaschinen so konstruiert wird, dass er in bezug auf seine eigenen Bauteile und die hergestellten Gegenstände möglichst homogen ist. In ein solches System kann der Mensch mit einbezogen werden.

In theoretischer Hinsicht ist das Problem der sich selbst reproduzierenden Systeme von einigen Mathematikern untersucht worden, wobei J.v.Neumann führend war. Als Modell dienen dabei zelluläre Automaten. Die einzelnen Zellen können auf verschiedene Zustände gebracht werden, die elementaren Bausteine für Schaltungen entsprechen. Es wird angenommen, dass am Anfang sich alle Zellen in einem Grundzustand befinden. Durch Impulse von aussen können nun die einzelnen Zellen einem Programm entsprechend in solche Zustände versetzt werden, dass sie ein datenverarbeitendes System, also einen Computer, bilden. Dieser Computer ist nun wiederum in der Lage, sich selbst in einem Nachbarfeld des zellulären Automaten neu zu konstruieren. Darüberhinaus ist es möglich, solche Anfangsmuster rechnender Schaltungen zu bilden, die stufenweise den Aufbau immer komplizierter Schaltungen also z.B. immer leistungsfähigerer Computer steuern können.

Auch diese Ideen werden heute noch wenig beachtet und mehr für interessante Spielereien gehalten. Das liegt daran, dass für einen praktischen Einsatz ausserordentlich grosse zelluläre Automaten benötigt würden. Die Situation kann sich aber in Kürze ändern. In dem Maße, wie die Miniaturisierung fortschreitet, gewinnen solche Verfahren vielleicht eines Tages praktische Bedeutung. Verschiedene Anzeichen deuten darauf hin. Die Schablonen für die Herstellung miniaturisierter Schaltkreise werden heute bereits auf Computern errechnet und über automatische Zeichengeräte aufgezeichnet. Die Selbstreproduktion bzw. Replikation von Computern durch Computer ist also bis zu einem gewissen Grade bereits Wirklichkeit.

Die Idee der Evolution von elementaren Mustern bis zu immer komplexeren Gebilden ist dabei wohl einer der faszinierendsten Aspekte der zukünftigen Datenverarbeitung bzw. Informatik. Hier kann der Mathematiker seinen vollen Scharfsinn einsetzen und von zunächst abstrakten zu praktisch einsetzbaren Systemen kommen. Das Prinzip der Keimzelle ist in mancher Hinsicht invers zur Idee des Axiomensystems, wobei folgender Unterschied besteht:

Ein Axiomensystem besteht aus einer Reihe von elementaren Sätzen, die ohne Beweis als gültig angenommen werden. Aufgrund eines Kalküls lassen sich andere Sätze daraus ableiten, bzw. es lassen sich die Sätze eines Lehrgebäudes z.B. der Euklidischen Geometrie darauf zurückführen. Der Prozess dieser Ableitungen und Zurückführungen ist jedoch nicht Gegenstand dieser Axiomensysteme sondern des damit arbeitenden Mathematikers. So gesehen hat ein Axiomensystem statischen Charakter. Die Axiome "gelten". Im Gegensatz dazu handelt es sich beim Evolutionsprozess aus einer Keimzelle heraus um ein dynamisches Prinzip. Die Gesetze für die Entfaltung aus dem verhältnismässig einfachen Anfangszustand heraus zu einem leistungsfähigen Computer, zu einem Organismus, zu einem mathematischen Lehrgebäude, zu einer Fertigungseinrichtung oder gar zu einem Kosmos sind integrierende Eigenschaften des Systems. Das ist aber auch der Gegensatz zwischen der statischen Auffassung des antiken (griechischen) Kulturkreises und unserer abendländischen faustischen Denkweise.

So sehe ich in der Theorie der Keimzelle eine Aufgabe von grösster Tragweite, die die Krönung der Theorie der Datenverarbeitung darstellen kann. Damit steht die Datenverarbeitung in bezug auf den wissenschaftlichen Rang keiner anderen Disziplin nach. Sie kann im Gegenteil eine zentrale Bedeutung bekommen und durch ihre Betrachtungsweise viele andere Gebiete von Forschung, Wissenschaft und Praxis befruchten. Sie wird zu einer völlig selbständigen Disziplin. Das ist es, was ich unter Emanzipation der Datenverarbeitung verstehe.

Wer diese Perspektiven vor Augen hat, der weiss, dass das Zeitalter des Computers und der Datenverarbeitung noch vor uns liegt, und so darf ich hoffen, dass der heutige Tag nicht den Abschluss einer Lebensarbeit, sondern den Beginn einer neuen Phase der Entwicklung bedeutet.

### **Emancipation of Data Processing**

Konrad ZUSE

Originally the computer development was initiated by engineers and scientists. Babbage, the real father of the computer was mathematician. Also the further developments in the USA, beginning approximately in the time of World War I, were effected by scientists and engineers like Stibitz, Aiken, Eckert, Mauchly, who had the aim to create efficient instruments for their own problems. I myself was civil engineer and from the beginning I had engineering problems in mind. So there was a close connection between calculating machines and mathematics. This does not only concern the problems to be solved by these machines but also the design and the theoretical basis for the construction of the machines themselves.

There are different conceptions possible. Looking at my own developments I can state with satisfaction that, for instance, the application of the mathematical logic to switching diagrams was very effective. That induced me to create a theory of general calculating based on the one hand on exact mathematical formulas, but on the other hand fully adjusted to practical tasks. In the forties, there was already a harmonious collaboration with representatives of mathematical logic. My mathematical cooperater, Mr. Lohmeyer, came from the school of the well-known logician Herrmann Scholz in Muenster. In spite of the very abstract opinion of logicians at that time he clearly realized the new possibilities for practical application.

Our development in Germany was completely based on practical problems. Furthermore, there was no connection between the developments in the USA and in Great Britain. In 1945, it was impossible for us to continue our work. So I concentrated on the problem of defining general algorithms. That led to the "Plankalkuel" (programming calculus) which we can label today as the first algorithm language.

Today we can investigate which connections existed between the development of calculating machines and mathematical theories apart from the mentioned own developments.

**Turing** published his book "On Computable Numbers".

**Shannon** published a paper on switching algebra (apart from one of Hansi Piesch).

**Stibitz** constructed a relay computer for complex numbers which was similar to the machines Z3 and Z4.

**Aiken** constructed several machines and developed some ideas in the direction of switching algebra.

**John v. Neumann** influenced the development especially in the first post-war period in the direction of stored-program-machines.

**Norbert Wiener** published general ideas under the name of Cybernetics for control problems.

Furthermore, the **Information Theory** and **Automata Theory** were developed by different scientists.

Today, after a sufficiently large interval, we may analyze these contributions. Meanwhile, Turing's ideas have gained great importance. The "Turing-computability" has become an important conception for the theorists. Of course, Turing's work is very abstract and hardly based on practi-

cal problems. Therefore, it scarcely influenced the computer development.

Today, the idea of the switching algebra is generally accepted. Shannon did not yet refer directly to the propositional calculus in his first publication. Nevertheless, today "and-elements", and "or-elements" are taken for granted.

I do not know to what extent Stibitz applied the mathematical logic as an instrument for the design of his machines.

John v. Neumann has strongly influenced the computer development. The idea of regarding the program as an object of the data processing is, of course, already anticipated in the "Plankalkuel". But J.v. Neumann constructed the first computer with stored program and calculating of addresses. Therefore, we are speaking of the "J.v. Neumann machine".

At that time Norbert Wiener's ideas created a sensation. But today we realize, that he himself merely influenced directly the computer development. For some time the computer science was understood as a section of Cybernetics, a conception not being generally accepted today.

The Information Theory was created already before the computer development as a means of telecommunication engineering. The principal aim is to get a neutral measure for the information, e.g. in order to compare the efficiency of different communication channels. In this theory the probability calculus, the calculating of the influence of troubles etc. are of great importance. The Information Theory, however, is not the theory of information processing. On one side, information is interpreted as to quantity, and on the other side as to quality. Unfortunately, the insufficient distinction of these two conceptions has caused much confusion.

Today Automate Theory belongs to the instruments of the Computer Science. Nevertheless, it is also rather abstract. It can help to understand some theoretical aspects. The practical development has scarcely been influenced by it.

The close connection of the computers with the mathematical science induced the mathematicians to regard the whole complex as their domain. That has many advantages but involves some dangers, too. The word "Informatik" which is used in German language for Computer-Science, obviously is derived of the terms information and automation. Information, however, is interpreted as mentioned before as to quality. The representatives of the "Informatik" (Computer Science) strongly tend to theoretical conceptions and formulations. The clearly defined conceptions of the information theory induce the mathematicians to use them. In the course of years, that led to the fact that the Informatik respectively the Computer Science has become a very abstract science whereby we should realize that theory and practice do not harmonize in every respect. No doubt, the Informatic could get a considerable independence and could emancipate to a certain degree from the other branches of science, especially the mathematics. But it did not succeed in integrating satisfactorily the whole complex of questions. Theoreticians and experts often speak a different language and have great difficulties in understanding each other. In my opinion this difference between theory and practice is not justified. It simply cannot be true that exactitude only is possible by abstract theoretical formulations.

At first sight, information processing and data processing seem to mean the same. The word information, however, sounds more scientifically. Nevertheless, we have already seen that there is a considerable difference between the quantitative conception of the Information Theory and the qualitative one of the "Informatik". In contrast to that, the word "data processing" is considerably less burdened. One speaks of the "information content" but not of the "data content". Mathematicians may say perhaps, that the term data is not clearly defined in comparison with the term

information. But the strong definitions of information are only good in the scope of the Information Theory and of its targets. It may be useful for a mathematician to put the question of the conservation of information in the different data processes. But one must not overrate this point of view. Mostly one gets by data processing new informations from old ones in the qualitative sense. Generally, information gets lost because only few data are often evaluated from plenty of data in the sense of Information Theory. Contrary to that, any amount of information can be produced by random numbers. The concepts of the Information Theory should only carefully be applied to data processing. This also refers to the conception of the "information flow".

Therefore, I suggest to go the following way: The conception of data or date is introduced as a fundamental element in the Data Processing Theory. The basis element is the bit. We do so without deducing decision between two alternatives. All other data structures are composed by bits. So we get an elementary statement:

"Data processing begins with the bit".

Data are objects of computing and all data have structures given by definitions. They have to be constructed in such a way that they can be clearly conceived as a whole or in their components. As the bit represents the elementary data structure, the operations with bits are elementary, too.

These ideas have induced me to develop the "Plankalkuel". It is the only algorithmic language, consequently designed corresponding to these points of view. Here, I can only illustrate this by some examples. All data have a structure. This allows several "values". Within programs they can be specified by names. Mostly the set of the possible values of a structure is finite. But, in principle, we can define structures with infinite numbers of values. The conception "potentially infinite" introduced in the mathematics is also good in this case.

In modern mathematics the set theory is generally applied with great success as an efficient instrument. A set can be defined in different manners, e.g. by predicates holding for all elements of the set. The direct definition is the listing of the elements. All these descriptions of sets are data. For listing we need the definition of data structure of the individual element and that of a list. A characteristic feature of data processing is: The elements of a list are lined up in a fixed order. One can define a set by such a list. The mathematical conception of the set, however, does not include an order of elements, i.e. originally, the same set can be described by different lists. All these lists have the same structure but every arrangement of the elements represents another date. One can now define an operation (program). "Comparison of two lists" with the result "the lists consist of the same elements". The lists are equivalent as to sets.

In this manner, we can represent the operations of the set theory by operations of the data processing. We can set up a self-contained list calculus corresponding to the set calculus. I already gave some guide-lines in the Plankalkuel. Although a set has normally no preferential order of the elements the mathematicians often use the term "ordered set". We can take the view that e.g. the set of the natural numbers lies between the conception "set" and "group". The definition of operations with certain characteristics on sets has led to the Group Theory which has proved to be very efficient in their various modulations. A list defines economically a set and the order relation of the elements. Infinite lists can be defined recursively like infinite sets or by a generative program.

In principle, there are many arbitrary variations to represent numbers by dates. A basic form is the binary system. All operations with numbers can be defined by programs. The transition from finite to infinite sets (both for natural and for rational numbers) can be done again by recursive definitions or programs.

The mathematician may object that the axioms for numerical operations hold, in principle, only for infinite sets. The practical construction of arithmetic units is always a compromise. From this

point of view the axiom-systems are often only an illusion. They only can be accomplished mentally by transition to dates of infinite structure. Concerning arithmetic calculations as data processing we need recursive definitions for the operations, too. By this way, the calculus of data processing may be more precise and instructive than traditional mathematical methods.

In computer-science we do not always understand by data structure the same. The interpretation of the Plankalkuel was not accepted. There are data structures defined by trees, allowing the access to every branch by a program; everything more requires a set of programs defined on such structures. Here the idea gets actual to extend the conceptions similar to the transition of the set theory to the group theory. In order to give an example we take the list of pairs which can be ordered according to the first or to the second elements of the pairs. Both lists represent the same set but different data. A neutral form would be the representation by a matrix well-known for the description of relations. But for long lists it is too clumsy. We can make general programs for the inversion of such lists of pairs. They only form a subset of the programs definable on pair lists.

I think, it may be better to use the concept of data-structure of the Plankalkuel and to speak of special program-sets for all further relations.

From this point of view the meaning of data processing could somewhat differ from that of informatics or computer science; but seen from a general scope there is no reason why distinguishing both conceptions. It is only important that the data processing or a corresponding conceived informatic emancipates from the sometimes too strong influence of theoretical mathematics. I already emphasized that this does not mean losing logic strength. On the contrary, I hope that some practical aspects can become more transparent.

Now I want to speak about some special perspectives in order to justify such an emancipation.

Even if the first computers were created by engineers and scientists for their own tasks, we can state nowadays that 80 to 90 % of all computers are applied for commercial and administrative affairs. As long as scientists concentrated on their own developments they were not taken very seriously by others. There were plenty of problems to be solved by computers. Nobody became unemployed. On the contrary, never before so many mathematicians were wanted.

Only the advancement in branches which embraced the entire economic and social life called man's attention. This is enforced by the introduction of the integrated switching circuits and the mini-computer. Nowadays nobody can avoid this development, but all these tasks taken over by computers are data handling processes. Therefore, we need a language which is understandable for all people involved. Finally, we proceed in the direction of the administration and control of the globe by means of gigantic data processing systems. There is still a lot to do. The possibilities to apply a computer not only for administration but also in the sense of artificial intelligence are scarcely recognized. We must not be encouraged by disappointments and setbacks in this field. The technology of integrated circuits offers new possibilities for associative data processing, which is a task for the following years.

Today theoretical physics is one of the most difficult and abstract fields in science. By enormous, formal efforts, mathematical means are applied in order to investigate the laws of the cosmos. I was pleased that last year we had in Berlin a congress, common for physics and computer science. The close connection between both fields was assumed only instinctively. Today computers are efficient instruments for the physicists; nevertheless, none of the physicists has in mind that the cosmos may be a gigantic data processing system. The suppositions for such a scope have still to be created. My own developments in the field of "Rechnender Baum" (computing cosmos) go in this direction. I had the occasion to simulate some models of this line representing cellular

automata with the computer at the University in Braunschweig with Professor Vollmar. But the results are not at all sensational, but have a more playful character. Nevertheless, we hope that one day new findings will result from these experiments.

I take the liberty of sketching some daring perspectives of that line. In the sense of a digitalized cosmos the quantum theory is only a step to overcome the concepts of continual spaces, differential equations etc. The physicists have strictly to adhere to the phenomenas and theories proved by experiments. Considering the Big Bang they assume that at that time our modern physical conceptions together with all physical quantities, elementary particles etc. existed already. Our knowledge of today does not allow another conclusion. But some time later perhaps one regards this point of view as naive as the idea of the scientists concerning the development of species before Darwin. Perhaps it will be necessary to assume a similar theory of evolution for the development of the cosmos, as in biology. Perhaps a complicated data processing in a limited space preceded the Big Bang in the sense of an enormous extension beginning at a distinct point in the development. Nature made a great many experiments. The electron survived. All that is only inventiveness, but I hope that new aspects like that of data processing one day will lead to fundamentally new theories.

Data processing is already successfully applied for automation. But we are only in the beginnings. It is still a long way to be able to control the development of a complete organism beginning with some kind of germ-cell by a program. Nature is superior with regard to the information stored in the DNA-chain. Finally, all that is also data processing.

In the evolution of biologic organisms the self-replication plays an important role. In this direction our technics and science are only in the beginnings, too. Our production methods differ from that of nature. Nevertheless, I am convinced that a revolutionary step is ahead of us. The self-reproducing systems were investigated by me as to the constructive aspects. I had in mind to construct a workshop which is able to replicate its components. This concept may be conceived more or less strongly. An immediately achievable aim would be the self-relying systems, without the claim for full self-reproduction. Here is only necessary that similar to the self-sufficient economy a set of machine tools is constructed so that it consists of the same type of parts which belong to the objects produced by them. In this sense the system is as homogeneous as possible. Man can be included in such a system. Theoretically, the problem of self-reproduction was analyzed by several mathematicians, whereby J.v.Neumann was prominent. Cellular automata serve as models. The individual cells can transfer to different states which correspond to elementary switching elements. One assumes that at the beginning all cells are in a fundamental state. By means of pulses from the exterior the individual cells can be transferred according to a program, so that they form a data processing system, i.e. a computer. This computer is capable of replicating in a neighbouring field of the cellular automata. Furthermore, it is possible to form patterns of a calculating diagram which can control stepwise the construction of more complicated diagrams, i.e. more efficient computers.

Also these ideas nowadays are considered more as interesting games. The reason is that extremely big cellular automata would be necessary for a practical application. Together with the progress in miniaturisation such methods may get important one day. Some aspects lead to such conclusions. Patterns for the manufacturing of miniaturized switching circuits are already calculated by computers and designed by automatic plotters. The self-replication of computers is already to a certain extent reality. The idea of the evolution of elementary pattern to more and more complex systems is one of the most fascinating aspects of the future data processing. Here the mathematicians can proceed with their full cleverness from abstract to practically applicable systems. The principle of the germ-cell is in some respect inverse to the idea of the axiom-systems whereby we find the following differences

An axiom-system consists of a set of elementary statements which are considered to hold without proof. By means of a calculus other statements may be derived or the statements of a field of science, e.g. the Euclidian geometry may be reduced to them.

The process of these derivations is not integrated in the axiom-system but is matter of the mathematician working with it. From this point of view an axiom-system has a static character. The axioms hold. In contrast to that the evolution process from a germ-cell has a dynamic character. The rules for the development from a relatively simple pattern to an efficient computer, to an organism, to a mathematical doctrine, to a production system even to a cosmos are integrated features of the system. That is the contrast between the static conception of the antique civilisation and our occidentally (faustischen) mentality.

In the theory of the germ-cell I see a task of great importance. So data processing may achieve the same scientific level as other scientific branches. Furthermore, it can get central importance and stimulate many other fields of research, science and practice. That is it what I understand by emancipation of data processing.

Seen from this perspective we realize that the computer and data processing aera is still ahead of us. I hope, that this day does not mean the end of a lifework but the beginning of a new phase of development.