Title: Looking Forward and Looking Back the rediscovery of burled ideas from the pioneer age of computers

Author(s): Konrad Zuse

Date: 1975

Published by: Konrad Zuse Internet Archive

Source: Document - ZIA ID: 0612

Looking Forward and Looking Back*

the rediscovery of burled ideas from the pioneer age of computers

Konrad Zuse

Lecture given at the
Workshop on Computer Architecture
Erlangen 22nd and 23rd May 1975

If we look back at the pioneer time in the development of computers, the first thing we notice is that at that time the contemporary distinction between hardware and software did not exist. The expression "computer architecture" fits in many ways that time better than the present one.

Initially, the central problem was one of construction. Babbage was a mathematician but he became an engineer in order to realise his ideas as a machine.

Also those who led the first developments of computers in the U.S.A. thirty or forty years ago, Aiken, Stibitz, and others were not specialists. They concerned themselves with purely technological matters as well as with logic design and programming.

It was the same in Germany. I myself was a civil engineer and hat to study mathematical logic for building computers. Also my friends of the pioneer time, Professor Billing and Professor Piloty, whom we can welcome here, were not specialised.

It ist difficult today to find a computer expert who is able, even to a limited extent, to deal knowledgeably with the wide range of these subjects. However, the very theme of this workshop, computer architectures, should remind us that an overall view is more important than ever, if we are to solve the problems we are faced with.

The following remarks are made from the point of view of my own work. It is apparent today that many old ideas which at that time were put aside as premature, can come into their own.

*ZuP 035/052. ZIA 0612. Version 2. Durchgesehen von R. Rojas, L. Scharf, G. Wagner.

In architecture one is aware of a variety of styles. They are in general based on the available methods of building. Wood, stone, concrete, steel, and other building materials lead to their own architectural style.

Computers can be regarded in a similar way. In computer components we have seen the development from mechanical to electromechanical technology, from valves and transistors to contemporary integrated circuits. Each of these technologies has its own style which influences the corresponding computer architecture.

Traditional calculating machines were used almost exclusively for numerical calculations in the decimal number system. They had as components wheels which acted as registers by representing numbers in discrete digital positions. Babbage´s work was still based completely on this style.

The recognition that, fundamentally, all computing operations could be based on operations on logical values (true or false) was an important advance. As a result the relay became the basic computer component, but rough calculations soon showed that many thousands of such relays would be necessary to build a usable computer. Especially the construction of the memory required a physically small and economically acceptable solution. As a consequence, the idea of a technology based on mechanical switching arose. This technology is today only of historical interest however, in looking back we can see several important characteristics, which can today gain importance for quite different technologies. The mechanical switching element simulated the functions of an electromechanical relay using movable plates, connecting rods and levers. It had the advantage over the electromechanical relay that it had a more compact construction. On the other hand it had the disadvantage that not only the logical circuits themselves had to be developed, but also the topological and spatial relation of the switching elements to each other. Here the neighbourhood relations between the individual elements played an important role. Transfers over larger distances led to complicated constructions using levers and had inherent limitations.

Today, we have a similar situation in the use of integrated circuit technology. The trend to implement, as far as practically possible, complete arithmetic and logical units on a single chip leads to densely packed arrangements, where the switching element which are logically connected to each other should also be, as far as possible, physically adjacent to each other. Connections from one chip to another are relatively expensive, and are to be avoided as far as possible. Also transfers over larger distances lead to problems, as at very high frequencies the velocity of light begin to have a limiting effect. Should one not look again at the old drawings of arithmetic units based on mechanical switching units, in order to give them a new life in an implementation based on modern technology?

Due to the slow speed of mechanical switching elements only parallel arithmetic units could be considered. Further, the idea of “superparallelism” arose. It is interesting to note that as late as about 1950 a manufacturer of punched card equipment still had not complete confidence in the electronic computers which were being developed at that time. He did not want to put all his money on one horse and gave us a contract to development a punched card calculator based on mechanical switching technology. We developed a system with a set of adders connected in series, through which the operations assigned to one punched card were processed. Thus, a series of arithmetic units were simultaniously occupied with the operations for several punched cards. This technique is exactly what is now known as pipelining.

At that time, we hoped to use this concept of adders working in parallel, to components for the relatively slow speed of the mechanical switching technology, compared to electronic technology and so to achieve an acceptable computing speed. In the meantime, the requirements for computing performance have increased enormously and an integrated circuit technology has been developed, which allows the concentration of large numbers of switching elements in a small space, whereby the spatial arrangement should as far as possible reflect the logical interconnections. Thus, we have another situation where an old and almost forgotten idea can remerge in modern dress.

An intermediate stage in the development is represented by relay computers based on an electromechanical technology. The use of relays led directly to the development of switching algebra. This was, indeed, a triumph for the formalism of the propositional calculus. As soon as one had grown accustomed to the idea that calculating machines need not remain limited to small desk calculating machines, and that whole rooms could be filled with cabinets containing relays, the path was clear to the construction of circuit of an arbitrary degree of complexity, in which the logical and arithmetic functions were largely implemented by hardware. In this way machines were developed with complete floating-point arithmetic for decimal and binary number systems. In which many special cases could be treated separately. The machines Z3, finished 1941, Z4, and others, fall into this category.

This highly developed switching technique was not directly continued, as the electronic technologies gradually matured and reliable electronic computers could be built.

At that time – about 1938 - 1945 – Schreyer and I still planned a direct transformation of the relay circuits into equivalent valve circuits. The problem of the development of the necessary basic components was then solved by Schreyer. However, we never got as far as actually building a complete computer. Also, the later developments showed that every technology involves its own computer architecture. The systematic transformation of relay circuits into valve circuits would have been very expensive. The very high speed of electronic circuitry led at first to a line of development which was opposed to superparallel systems made of mechanical switching elements, i.e. systems involving pipelining. The adder which was still parallel in relay machines where each binary position was allocated to a separate switching unit, could be reduced to a serial adder in an electronic computer. In the latter the binary digits which make up the operands are processed one at a time by a single bit adder. This had further consequences for the transfer of data which could now be carried out over a single wire. In this way the intrinsically higher costs of the electronic components could be compensated for. Also several serially working memories were constructed according to this principle. For a while, the drum was the ideal store both technically and economically for medium size computers.

Later there appeared memories which were again parallel, similar to the mechanical memories of the pioneer time and the relay stores which contained a large number of relays. Among these ferrite core memories were the most important. The highly developed logical circuit technology of the relay devices was abandoned and the corresponding tasks were carried out by software. So came about machines like the Z22, which had the advantage that the user could build up his own instruction code, and could by flexible programming improve performance in comparison to relay machines, also from a logical point of view. The present level of integrated circuit technology makes it again possible to construct arithmetic units with complete floating-point hardware. This hardware is again able to perform tasks which, in a sense, hat to be performed by software during the intermediate period. Thus, a certain similarity is noticeable between the structure of modern computers and those of the pioneer time.

Also typical for the development of computer architecture is the concept of cellular automata. The idea of connecting many parallel arithmetic or logical units together in a lattice is intrinsically trivial. Numerical methods for the solution of partial differential equations which operate on the elements of an array arranged as a lattice, existed already before the development of computers as they are known today. However, it was at that time extraordinarily laborious to carry out such calculations. It would have been absurd in the initial period of the development of computers to actually construct cellular automata. At that time one had good reasons to be thankful when one could build individual arithmetic units which were reliable and economically viable.

As soon as the electronic computers of the beginning of the fifties became known, the idea naturally occured to the meteorologists to use them for weather forecasting.

Despite their relatively high speed in comparison to traditional computers, the computers of the early fifties were not fast enough for weather forecasting. The weather changed faster than the computers could forecast it. In this situation the idea of the array computer emerged as a compromise. At that time the drum was the only available storage medium with sufficient reliability and capacity, which was economically viable. An obvious idea was to use it as directly as possible as a cellular automaton. A number of parallel tracks on the drum could be regarded as an array, which could be processed during a single revolution of the drum. Thus, two such arrays could be added to form a third array in a single operation. Similar algorithms could be developed for multiplication, shifting, etc.

This array computer was described in various patent applications. It remained, however, only a paper computer. The reason is to be found in the danger to which every technological development is exposed. The good idea was conquered by a better one.

The array computer was suited to the technology of the time round 1956. At that time it was sensible and economically sound. However, important advances occurred in the general lever of computer development. The speed of the arithmetic units was significantly increased and large capacity care stores could be built economically. As a result it became possible to use fast general purpose computers not containing special hardware, to solve partial differential aquations. The routine use of computers in weather forecasting could begin.

Since then, technology has advanced again. The original idea of a cellular automata is today no longer utopian. New technological developments encourage us once more to look again at old ideas. Particularly, for the implementation of cellular automata a number of possibilities present themselves. The aim is here not only to construct a tool for the solution of partial differential equations. The concept of a computer automata can also be a basis for fundamentally new theories. A digitalisation of space can also stimulate new advances in theoretical physics, for instance simulation in three-dimensional space. Thus, we come back once again to the pioneering age, when hardware and software both could find a place in the minds of some of the pioneers. Especially, the cellular automaton can only be developed to a successful conclusion when the technological possibilities and the logical concepts are closely harmonised with each other.

I would like to consider the subject of algorithmic languages. The first computers of the pioneering time, i.e. those which were available about 1945, did not require a complicated programming technique. A special situation arose in Germany. After 1945 we were at first out off from the rest of the world. It was inconceivable that we should develop new hardware for years to come. This isolation allowed me to concentrate my attention completely on theoretical problems. So I was able in a small and remote alpine village, to write what appears to be the world´s first algorithmic language, the "Plankalkül". This work was at first put on one side. Also, when the development of programming languages gained momentum between 1950 und 1960 other approaches were adopted. The first programming languages were of necessity based on the then current computer technology, while the "Plankalkül" attempted to cover the whole spectrum of possible calculations.

The further development of the field of programming languages has not been very happy. We can today speak of a definite crisis in software. A confusion of languages, all too reminiscent of the tower of Babel, characterises the present situation. To try to create at least a partial order out of the present chaos, the present trend is towards structured programming. If one now looks again at the Plankalkül, one sees that it is constructed according to the principle of structured programming. It contains, among other things, the following basic and simple concepts.

1. Construction of all data structures on the basis of the single bit.
2. Completely modular structure of programs and subroutines with clear specification of input and output values.
3. The use of the logical operations of the propositional calculus.

The Plankalkül was developed as a purely logical algorithmic language, i.e. contains no elements which are immediately related to the implementation, such as input and output statements, storage allocation instructions, etc. It can, however, be implemented without undue difficulty. Today we are at the transition to new and unconventional computer architectures. A series of tasks which up till now had to be left to programming, compilers, operating systems, and run time systems, could in the future be largely performed by hardware. Perhaps, we are not so far away from the situation in which computers are more or less directly controlled by logical algorithmic languages, such as the „Plankalkül".

The logical operations of the predicate calculus are well suited to writing programs for computers with associative stores. Also in this context, we find that ideas from the pioneer time of computers take on a new life. Anyone who considers all these ideas carefully recognises that it will be a long time before the development of computers will reach a period of tranquillity.

Perhaps we stand at the threshold of a new age in which hardware and software in harmonious relationship will together enable us to find new solutions to our problems.

Hünfeld, April 25, 1975
K. Zuse