CompSci 790 (History) – Assignment 2

On Konrad Zuse's Developments

Dominik Schultes

2. June 2004

Abstract

Konrad Zuse, a German engineer, did pioneer work when he constructed several program-controlled calculators. In this essay, his developments between 1936 and 1945 are described. In Section 1, the basic concepts of Zuse's machines, such as binary and floating-point arithmetic, are explained. These principles have been implemented in Zuse's first four general purpose machines, the Z 1, Z 2, Z 3, and Z 4, which are examined in Section 2. Section 3 concludes with some remarks on Zuse's achievements.

1 The Basic Concepts of Zuse's Machines

At the beginning of the 1930s, Konrad Zuse studied civil engineering at the Technical University Berlin-Charlottenburg. As a student he had to perform many extensive calculations in the field of statics. These calculations initiated Zuse's thoughts on the automation of such computations. His goal was the development of a machine that could perform complex, schematic calculations automatically and that was adapted to the particular needs of engineers and scientists.

One of the most important properties of his machine is the fact that it is **program-controlled**[1]. The advantage is that a program has to be written only once and can then be used for an arbitrary amount of instances of the same problem. This situation applied to Zuse's calculations that he had to do at university: the same type of calculations repeated again and again, but each time with different numbers. Obviously, the concept of a machine that is controlled by a program has a great impact on the error rate of such computations. As soon as the program has been specified correctly and the machine works reliable, no further errors can occur, while there is the permanent jeopardy of a mistake when a human – even a very skilled one – performs the same calculation again and again. [5]

Zuse realized that even very complicated numerical computations could be expressed by such a program if they fulfilled two preconditions:

1. It must be possible to reduce them to basic operations.
2. They must be expanded explicitly.

If these conditions are satisfied, a program can be constructed in the following way: all variables (input, temporary, and output) that appear are enumerated. In todays terminology, these numbers correspond with the addresses of the registers [9]. Then, the program consists of a sequence of equations; each equation consists of the addresses of one or two operands, an operation and an address for the (temporary or final) result.

The hardware that can deal with these programs consists of the following components:

1. An *arithmetic logic unit* (ALU) that performs the actual calculations. After the operands have been set and the basic operation has been selected, the ALU performs the operation automatically.

[1] Zuse used the term "Rechenplan" [calculation plan] instead of "program".

2. A *register array* (in other words, *memory*) that stores the input and temporary data of a calculation. It consists of hundreds of cells – each cell can store one number. Each cell can be connected with the ALU in order to exchange the data, i.e., first the ALU reads the input from the memory, and then it writes the result back.

3. A *control unit* that controls the other components and the interaction between them according to the program flow. **The program is encoded on a punched card.** A punch card reader samples the card and interprets the instructions so that they can be executed. The storage of a program on a punch card has a big advantage over the "programming" of a machine by setting up switches and wire straps: after a program A has been executed, another program B can be run and, after that, program A can be used again without the need of programming it again.

4. A *user interface*, namely a control panel and a device that allows the user to read the results.

This **subdivision into arithmetic logic unit, memory and control unit**, which had been proposed by Charles Babbage for the first time, corresponds with the basics of computer architecture down to the present day.

When complex computations are performed, most values are only temporary results and stay inside the machine so that the user does not encounter them. Only the input data and the final result is given resp. read by the user and therefore only these values have to be in a representation that is convenient for the human user. However, internally a representation should be used that is convenient for the machine. Therefore, Zuse decided to use **binary arithmetic** because it allows a simple form of computation both from a mathematical and from a engineering point of view [6]. The main advantage is that simple components can be used that have the property that they are in either one of two states. However, for the convenience of the user, Zuse provided input and output facilities in decimal representation and the decimal-to-binary conversion and vice versa was performed automatically by the machine. It is remarkable that Zuse combined the representation that is convenient for the human user and the representation that is convenient for the machine. In many cases, developers have either a very human or a very technical point of view, and in the latter case they do not really bother to regard the convenience of the user. Zuse contrived to unite both fronts.

Zuse observed that **all operations of binary arithmetic can be reduced to the basic logic operations AND, OR, and NOT** [8]. This enables the application of the relay technology on a grand scale. In addition to the common electromagnetic relays, Zuse considered mechanical relays because at that time relays were quite expensive and much space was required for a machine that consisted of a large amount of relays – for Zuse, both money and space were limited. The mechanical relays designed by Zuse had the same functionality as electromagnetic relays, but had the advantage that they could be constructed manually and the material costs could be kept low. Furthermore, they were comparably small.

Zuse noticed that the order of magnitude of numbers varied in a huge range when scientific or technical computations were performed [3]. Calculations where a number of the order of 10^6 and another number of the order of 10^{-6} appear in the same equation occur frequently. Zuse remarked that usually one of the following two solutions was chosen when a calculating machine was built at that time. Either machines with very many digits were built or the user had to calculate the order of magnitude of the result in advance in order to adapt the order of magnitude of the input so that the available digits of the machine could be used in an optimal way. The former solution has the disadvantage that the constructional effort is very high, while only a small fraction of the digits of the final result is relevant. The latter one has the disadvantage, that the user has to know something about the result, before it has been computed. Therefore, Zuse decided to implement another solution, namely **floating-point arithmetic**[2]. Each number $y = 2^a \cdot b$ is represented by an integer exponent a and the mantissa b. The amount of digits of the mantissa should correspond with the arithmetic precision of the machine so that there are as many digits as actually used. Due to the flexible exponent, the order of magnitude can vary in a wide range. Again, Zuse distinguished between the internal representation that is convenient for

[2]Zuse used the term "halblogarithmische Form" [semi-logarithmic representation].

the machine and the external representation that is convenient for the human user. The user could feed a number with an arbitrarily placed decimal point into the machine and the conversion to the floating-point representation was done automatically. Vice versa, the result in the floating-point representation was converted so that the output consisted of a number with the correctly placed decimal point.

In addition to a special representation of the number 0, which cannot be represented directly, Zuse introduced three **special values** *infinity*, *not defined*, and *imaginary*. Division by zero results in *infinity*; $0/0$, $0 \cdot \infty$, and $\infty - \infty$ leads to *not defined*; the root of a negative number yields *imaginary*. This treatment of special cases has the advantage that the user can rely on the results of the machine. Nowadays, the programming language Java uses similar concepts like special values *NaN* [Not-a-Number], *positive infinity* and *negative infinity*, but there are still programming languages that are, in this respect, less advanced than Zuse's concept.

Zuse's machines supported several basic operations: addition, subtraction, multiplication, division, root extraction, squaring, calculation of the reciprocal and of the absolute value, multiplication by π, 10, 2 (left shift) and 1/2 (right shift), determination of the minimum and the maximum of two values, and an operation called *Fpos(a)* that returns 0 if the operand a is negative, and a otherwise. [7]

2 Z 1 – Z 4

In Zuse's first calculating machine, the **Z 1**, all of the above mentioned basic concepts were implemented; Zuse described them in his patent application on December 21st, 1936 [4]. The construction took place between 1936 and 1938. For lack of alternatives, Zuse worked with some friends in his parents' living room (Figure 1). Only mechanical components were used, i.e., both

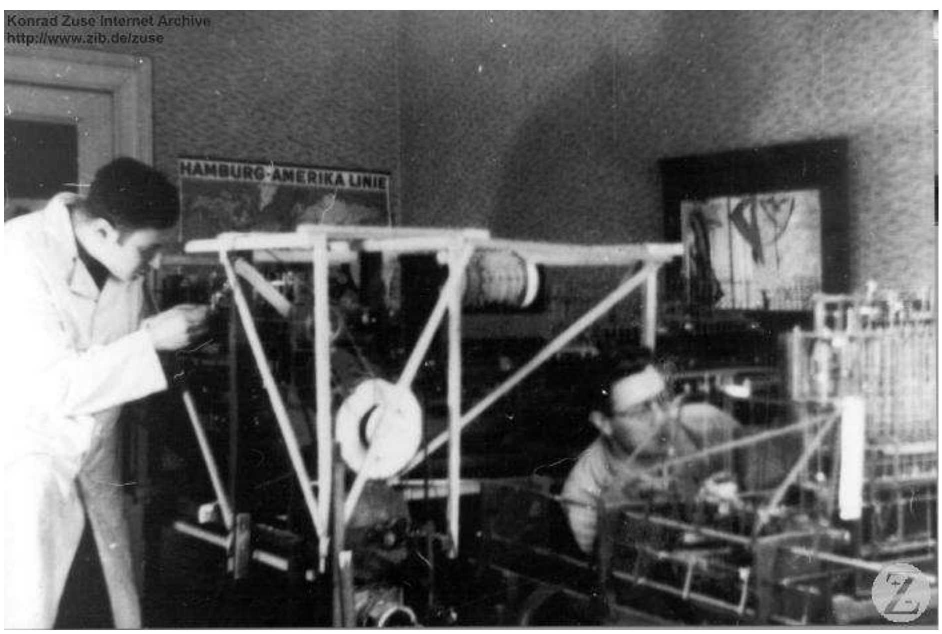

Figure 1: Workshop in Berlin, in the parental flat, H. Schreyer (l.) and K. Zuse, 1936 [12]

the memory and the arithmetic logic unit consisted of mechanical relays due to the advantages mentioned in Section 1. The parts were constructed manually using fretsaws. The original Z 1 including all records was destroyed during an air raid. Between 1987 and 1989 Zuse reconstructed the Z 1 from memory. It is on display at the German Museum of Technology Berlin.

The Z 1 did not work reliably; the components performed satisfactorily, but the interaction between the different components was not proper [11, p. 61]. Therefore, Zuse decided to test if electromagnetic relays could lead to better results. In 1939, he completed the **Z 2**. The storage still consisted of mechanical parts, but the arithmetic logic unit was built using electromagnetic relays. In contrast to his other machines, Zuse used fixed-point arithmetic in the Z 2. The Z 2 worked properly exactly once [11, p. 69], namely during a demonstration at the German Research Institute for Aviation [Deutsche Versuchsanstalt für Luftfahrt (DVL)]. On principle Zuse's design was fine, but the fact that he had used second-hand relays resulted in the unreliability of the Z 2. However, the short success story of the Z 2 encouraged the DVL to support Zuse during his further work. In return, Zuse developed two special machines that were used for measurements of wings and tail units of missiles and the computation of the optimal settings. [10]

After the successful test run of the Z 2, Zuse started to build the **Z 3** using electromagnetic relays for both the arithmetic logic unit and the storage. On May 12th, 1941 the first successful demonstration of the Z 3 took place in Berlin. It was the world's first fully operative program-controlled *calculator*, but it was not a *computer* from today's point of view as the conditional branch was not supported, although Zuse was aware of this type of instruction as he had implemented it in the microprograms for the floating-point arithmetic.[3] However, R. Rojas proved in [1] that in fact a single program loop containing only arithmetic instructions can simulate any Turing machine whose tape is of a given finite size. This means that conditional branching and indirect addressing – features that were missing in the Z 3 – can be simulated by purely arithmetic means that were supported by the Z 3. Of course, the Z 3 was less powerful than a Turing machine, which consists of an infinite tape, but it was, at least in principle, as universal as today's computers, which have a bounded addressing space because nobody can construct an infinite tape.

In 1944, the Z 3 was destroyed by bombs. In 1960, a reproduction was constructed, which is on display at the German Museum in Munich (Figure 2).

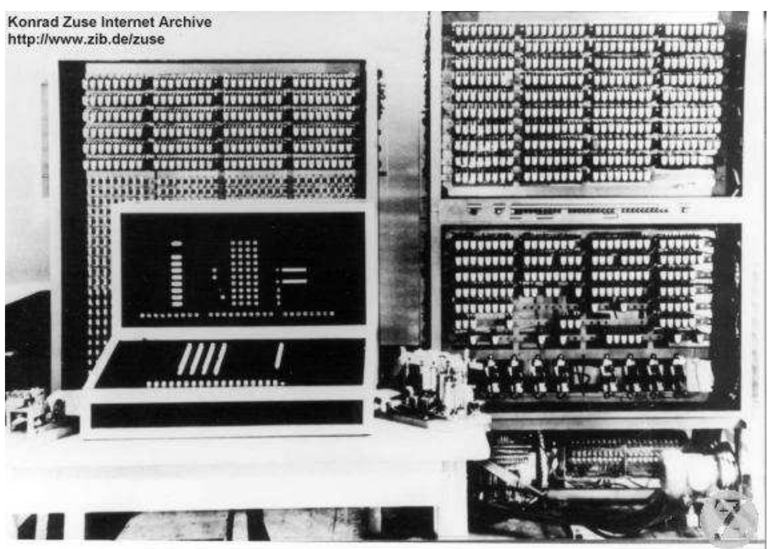

Figure 2: The Z 3, reproduction, German Museum in Munich, 1960 [12]

The **Z 4** was an enhancement of the Z 3, primarily with respect to the computation speed. Furthermore, Zuse added a device that could be used to create programs. This device was quite user-friendly so that the programming of the Z 4 could be learnt in less than three hours. Besides, it could be used to correct and to copy programs. In contrast to the Z 4, the Z 3 had been "programmed" by a manual puncher [11, p. 71]. In 1945, the Z 4 could be saved by taking it from Berlin to Göttingen where it was started up successfully [11, p. 83]. Then, the Z 4 was hidden in Hinterstein, a small village in the German Alps [11, p. 85]. Between 1950 and 1955, the Z 4 was operated by the ETH Zürich where it ran day and night. Since 1957, it is exhibited at the German Museum in Munich. [2]

[3] Most German texts state that the Z 3 was the first "Rechner". ("Rechner" is the German term for calculator and for computer.)

Figure 3: Konrad Zuse and the Z 4, 1948 [12]

3 Zuse's achievements

When assessing Konrad Zuse's achievements, the following aspects should be considered:

- Zuse developed many of the basic concepts that are fundamental for computer architecture till this day: control by programs that are reusable, subdivision into arithmetic logic unit, memory and control unit, binary and floating-point arithmetic with special values for exceptional cases. Furthermore, he realized that all operations of binary arithmetic can be reduced to the basic logic operations AND, OR, and NOT.

  Some of these concepts had already been described a long time before Zuse's developments, for example by Charles Babbage. However, Zuse was not aware of the work of his predecessors when he constructed the Z 1 – Z 4 so that he evolved these principles on his own. [11, p. 8]

- Zuse implemented his ideas determinedly. He used both existing and newly created components. He made progress step by step and did not make the mistake to let his ideas hurry ahead too far, but he always kept his eye on the feasible things.

- Zuse had not only a keen sense of machines, but he also realized the importance of a user-friendly design so that he always tried to unite these two worlds.

- Zuse worked under the worst conditions. First of all, the funding of his work was quite difficult. He had no private assets at his disposal – his father was a Prussian post office clerk [11, p. 11]. Nevertheless, most of his work had to be financed privately because the German government did not recognize the potential of Zuse's developments. In [11, p. 65, translated from German], Zuse gives an example for this ignorance: he had applied for leave from military service in order to continue his work on his calculating machines that could be used – among other things – for the computation of airplanes. During a conversation, his superior stated: "Actually, I don't understand the whole thing, what does 'computation of airplanes' mean? The German Air Force is faultless, where is the need for further computations?"

  Only the construction of the Z 3 and Z 4 was partly supported by public funds. The fact that Zuse had to use second-hand components caused many problems.

  Furthermore, Zuse's work was over and over again interrupted or disturbed by the war. He was drafted several times and only with the help of influential friends he was able to get a leave and avoid further calls. The Z 1 – Z 3 had been completely destroyed during air raids, including all records. The Z 4 had to be moved three times in Berlin, before it was transported through almost the entire country during the sunset of the war. Still, Zuse has never given up his developments. In 1945, he concludes in [7, translated from German] "Although the conditions in Germany are suppressive, we are bound and determined to continue the development of the calculating machines, whose current state is only the first step of an extensive development that can lead to results that we cannot anticipate today."

List of Figures

References

[1] R. Rojas. How to make Zuse's Z3 a universal computer. *IEEE Annals of the History of Computing*, 20(3):51–54, 1998.

[2] Horst Zuse. Konrad Zuse und seine Rechner. *Technical University Berlin, http://irb.cs.tu-berlin.de/~zuse/Konrad_Zuse/de/*, 2000.

[3] Konrad Zuse. Die Rechenmaschinen des Ingenieurs. *Konrad Zuse Internet Archiv, Zuse Institute Berlin, http://www.zib.de/zuse*, ZIANr 0234, ZuP 009/001, 1936.

[4] Konrad Zuse. Patentanmeldung Z23624 (Z1) "Rechenmaschine". *Konrad Zuse Internet Archiv, Zuse Institute Berlin, http://www.zib.de/zuse*, ZIANr 0177, ZuP 005/012, 1936.

[5] Konrad Zuse. Verfahren zur selbsttätigen Durchführung von Rechnungen mit Hilfe von Rechenmaschinen. *Konrad Zuse Internet Archiv, Zuse Institute Berlin, http://www.zib.de/zuse*, ZIANr 0176, ZuP 005/021, 1936.

[6] Konrad Zuse. Patentanmeldung Z391. *Konrad Zuse Internet Archiv, Zuse Institute Berlin, http://www.zib.de/zuse*, ZIANr 0229, ZuP 005/011, 1941.

[7] Konrad Zuse. Bericht über meine Rechengeräte. *Konrad Zuse Internet Archiv, Zuse Institute Berlin, http://www.zib.de/zuse*, ZIANr 0341, ZuP 010/010, 1945.

[8] Konrad Zuse. Ein neues Rechengerät für technische und wissenschaftliche Rechnungen. *Konrad Zuse Internet Archiv, Zuse Institute Berlin, http://www.zib.de/zuse*, ZIANr 0070, ZuP 037/007, 1948.

[9] Konrad Zuse. Entwicklungslinien einer Rechengeräte-Entwicklung von der Mechanik zur Elektronik. *Konrad Zuse Internet Archiv, Zuse Institute Berlin, http://www.zib.de/zuse*, ZIANr 0086, ZuP 005/001, 1960.

[10] Konrad Zuse. Vor den Erfolg haben die Götter den Schweiß gesetzt. *Konrad Zuse Internet Archiv, Zuse Institute Berlin, http://www.zib.de/zuse*, ZIANr 0028, ZuP 039/001, 1961.

[11] Konrad Zuse. Der Computer mein Lebenswerk (1st draft of Zuse's autobiography). *Konrad Zuse Internet Archiv, Zuse Institute Berlin, http://www.zib.de/zuse*, ZIANr 0281, ZuP 030/004, 1968.

[12] Konrad Zuse Internet Archiv, Zuse Institute Berlin, http://www.zib.de/zuse.