

Anmerkungen zum John von Neumann Rechner

Horst Zuse

TU-Berlin
Franklinstraße 28/29
10587 Berlin
Germany
Phone + Fax: +49-30-881 59 88
E-mail: zuse@cs.tu-berlin.de
Internet: <http://www.cs.tu-berlin.de/~zuse>

Keywords

John von Neumann; Konrad Zuse; John Mauchly; John von Neumann Rechner; Zuse-Maschinen Z1, Z2, Z3, Z4; Plankalkül; Stored-Program-Computer; Starre Rechenpläne; Freie Rechenpläne; Plankalkül; Turing-Maschine, Speicherprogrammierter Rechner; Universalrechner.

Abstrakt

Die Begriffe Neumann'scher Rechner oder Neumann'sche Rechnerarchitektur sind heutzutage in fast jedem Lehrbuch zu finden. Es wirkt argumentiert, daß die Neumann'sche Rechnerarchitektur diejenige des modernen Computers sei. Die Begriffe speicherprogrammierter Rechner und Neumann'scher Rechner werden dabei synonym verwendet. Es stellen sich aber zwei Fragen: Wie können die frühen Rechner ABC, COLOSSUS, ENIAC, MARK I und Konrad Zuses Z1-Z4 bezüglich des Neumann'schen Rechnertyp eingeordnet werden und ist der von Neumann Rechner tatsächlich ein neuer Rechnertyp?

1 Einleitung

In dem Artikel: *Heike Stach: Als Rechner zu abstrakten Maschinen wurden /STAC98/*, der 1998 in dieser Zeitschrift abgedruckt wurde, wird der Artikel: *John von Neumann: First Draft of a Report on the EDVAC /NEUM45/* diskutiert. Aus der Besprechung des Artikel geht aus unserer Sicht nicht klar hervor, ob es sich um eine Bewertung der Beiträge John von Neumann's zum modernen Computer handelt, es beabsichtigt ist, den State-of-Art in der Computerentwicklung um das Jahr 1945 darzustellen, oder ob es eine historische Aufbereitung der Entwicklung des Computers ist, wobei aus der Sicht von Heike Stach John von Neumann die entscheidenden Beiträge eingebracht hat? Dieser Artikel veranlaßte uns, die Arbeiten von John von Neumann, Burks, Goldstine und Mauchly kritisch zu analysieren und in diesem Artikel zu diskutieren.

Die Begriffe Neumann'scher Rechner oder Neumann'sche Rechnerarchitektur sind heutzutage in fast jedem Lehrbuch zu finden. Es wird argumentiert, daß die Neumann'sche

Rechnerarchitektur diejenige des modernen Computers sei. Die Begriffe speicherprogrammierter Rechner und Neumann'scher Rechner werden dabei synonym verwendet. Es stellen sich aber zwei Fragen: Wie können die frühen Rechner ABC, COLOSSUS, ENIAC, MARK I und Konrad Zuses Z1-Z4 bezüglich des von Neumann Rechnertyps eingeordnet werden und ist der von Neumann Rechner tatsächlich ein neuer Rechnertyp?

In Rojas /ROJA97/ ist auf Seite 62 ist zu lesen: *Als Abschluß für einen Vortrag über die Erfindung des Computers habe ich einmal drei Namen auf die letzte Folie geschrieben, nämlich Babbage, Zuse und von Neumann. Die Folie habe ich Konrad Zuse 1993 gezeigt, da ich naiverweise dachte, er würde sich freuen, sich in so illustrierter Gesellschaft zu sehen: Zuse fragte aber nur: Was hat von Neumann da zu suchen?* Es stellt sich die Frage, wie Konrad Zuse zu dieser Antwort kam, wo doch in fast jedem Buch über Rechnerarchitekturen der Neumann'sche Rechnertyp mit seinen Eigenschaften erwähnt wird. Hat Konrad Zuse hier nicht präzise genug zwischen (s)einer programmgesteuerten Rechenmaschine und dem modernen Computer unterschieden oder gibt es berechtigte Zweifel bzw. eine Mißinterpretation an den Beiträgen von Neumann's zur Computerentwicklung?

Unser Beitrag ist eine wissenschaftliche Bewertung des Beitrags John von Neumann's zur Computerentwicklung. Es ist keine Frage, daß John von Neumann ein hervorragender Wissenschaftler und Mathematiker war. Es geht hier nicht um die Bewertung dieser außerordentlichen Leistungen. Es ist aber unsere Sicht, daß die Leistungen John von Neumann's auf dem Gebiet der Computerentwicklung kritisch mit denen anderer Computerpioniere verglichen werden müssen. Es muß klar herausgearbeitet werden, welche logischen Konzepte von Rechenmaschinen (Computer), die von John von Neumann um 1945/46 diskutiert bzw. postuliert wurden, schon von anderen Computerpionieren vor John von Neumann diskutiert bzw. sogar implementiert wurden. Als Grundlage für unsere Bewertung der Leistungen John von Neumann's auf dem Gebiet der Computerentwicklung beziehen wir uns u.a. auf die Artikel /NEUM45/, /BURK46/, /MAUC47/, /ROJA98/, /BAUE98/ und auf die uns vorliegenden Artikel von Konrad Zuse zwischen 1936 und 1945.

Der Artikel ist wie folgt gegliedert: In Kapitel 2 werden frühe Rechner charakterisiert, in Kapitel 3 werden die Architektur der Maschinen Z1-Z4 von Zuse und die Architektur des von Neumann Rechners vorgestellt. Da die Architektur von Zuses Rechnern Z1-Z4 denen moderner Computer (Neumann Rechner) am ähnlichsten ist, werden Zuses Beiträge und Sichten zur Programmierung dieser und geplanter Maschinen erläutert. Im Kapitel 4 werden die drei wichtigsten Artikel /NEUM45/, /BURK46/, /MAUC47/ zum speicherprogrammierbaren Computer diskutiert, in Kapitel 5 wird der Neumann'sche Rechnertyp bewertet, in Kapitel 6 folgt die Zusammenfassung und in Kapitel 7 folgt ein Literaturverzeichnis.

2 Frühe Rechner

Bevor wir die Leistungen von John von Neumann auf dem Gebiet der Computerentwicklung einer kritischen Bewertung unterziehen und mit den Entwicklungen von Babbage und besonders von Konrad Zuse vergleichen, soll die gängige Definition des Computers von Ceruzzi gegeben werden /BAUE98/: *Before 1945 a person who did calculations. After 1945 a machine capable of the four operations of arithmetic, automatic storage and retrieval of intermediate results, and automatic input and output, all directed by a control unit. The modern definition is a machine which can manipulate symbolic information in any*

combination or way one desires, and which contains an internally stored program, which the machine may also manipulate if desired.

Wir stellen fest, daß die Definition des Computers sich über die Jahre geändert hat. Wir wollen nun die wichtigsten historischen Rechner vor 1945/46 kurz vorstellen:

- Charles Babbage (1792-1871) entwarf zwei Rechenmaschinen, die *Difference Machine* (1823) und die *Analytical Engine* (1834). Die Maschinen wurden niemals fertiggestellt. Das lag nicht an Entwurfsfehlern, sondern an der mangelnden Präzision der Feinmechanik zu dieser Zeit. Beide Maschinen basierten auf der Dezimalarithmetik. Babbage formulierte auch die ersten Ideen der Programmierung und kannte mit der *Analytical Machine* das Konzept von zwei Operationseinheiten¹ (*store* und *mill*) in der Maschine. Babbage adressierte den Speicher dadurch, daß seine Lochkarten für jede Speicherzelle eine gesonderte Lochposition hatten. Babbage kannte auch Sprungbefehle zum Verlassen des linearen Befehlsflusses.
- Der bis 1942 von Atanasoff gebaute Rechner ABC war ein nicht programmierbarer Spezialrechner in Röhrentechnik und dem Binärprinzip (Integerrechnung), welcher als Prototyp des Parallelrechners angesehen werden kann.
- Erst 1944 vollendete Aiken die MARK I, die noch ein dezimales Rechenwerk verwendete und die Trennung von Speicher, Steuereinheit und Rechenwerk nicht kannte. Die MARK I war frei programmierbar²
- Die 1945/46 fertiggestellte ENIAC von Eckert und Mauchly in den USA mit ihren ca. 18000 Röhren war nicht frei programmierbar und arbeitete mit einem Dezimalrechenwerk. Die Steuerung der Maschine (Programmierung) wurde durch das Setzen von Hunderten von Drehschaltern und das Stecken von Kabelverbindungen erreicht.
- Die in England von 1943-1945 gebauten zehn COLOSSUS-Rechner waren wiederum Spezialrechner mit Röhren im Binärprinzip, sie waren nicht frei programmierbar und wurden erfolgreich zur Entschlüsselung von Funksprüchen der deutschen Wehrmacht eingesetzt. Auf Befehl Churchills wurden alle Maschinen und Unterlagen 1946 zerstört.
- Konrad Zuses Rechner Z1 (1938) und Z3 (1941) kannten die strikte Trennung von Speicher, Steuereinheit, Rechenwerk (Gleitkommaarithmetik) und Ein- / Ausgabeeinheiten, waren frei programmierbar, verfügten im Gegensatz zur Dezimalarithmetik bei Babbage über ein Rechenwerk für binäre Gleitkommazahlen mit einer arithmetischen Ausnahmebehandlung, und basierten auf der binären Schaltungslogik (UND, ODER, NEGATION). Der Rechner Z3 war 1941 funktionsfähig. Während Babbage den Speicher dadurch adressierte, daß seine Lochkarten für jede Speicherzelle eine gesonderte Lochposition hatten, konnte Zuse den Speicher frei adressieren. Dafür konnten Zuses Rechner den sequentiellen Programmfluß nicht durch einen Sprungbefehl verlassen, es sei denn die Ausnahmebedingung $x/0$ wurde ermittelt und das Programm stoppte.

¹ Unter Operationseinheiten werden die Komponenten eines Computers verstanden: Kontrolleinheit zur Interpretation der Befehle und der Steuerung der gesamten Maschine, einem Speicher, einer arithmetischen Einheit, und den Ein- und Ausgabeeinheiten. Babbage allerdings trennte nur *the store* (Speicher) und *the mill* (Arithmetische Einheit: nur Additionen waren vorgesehen). Ein- und Ausgabeeinheiten waren nicht geplant bzw. vorhanden.

² Unter freier Programmierbarkeit wird die Fähigkeit eines Rechners verstanden, eine beliebige (zulässige) Folge von Befehlen durch das Steuerwerk zu interpretieren und damit das Rechenwerk, den Speicher oder die Aus- und Eingabeeinheiten zu steuern. Die Befehle sollen dabei auf einem Speichermedium vorliegen, welches die Maschine lesen kann.

Aus der obigen Aufstellung und der Untersuchungen von Rojas /ROJA98/ ergibt sich, daß die Rechner Z1-Z4 von Konrad Zuse dem Neumann'schen Rechner am ähnlichsten sind (siehe unten), vor den Veröffentlichungen von John von Neumann /NEUM45/, Burks et al. /BURK46/ und Mauchly /MAUC47/ konstruiert wurden und mit der Z3 im Jahr 1941 der erste Rechner funktionsfähig war. Um die Unterschiede zwischen Zuses Maschinen Z1, Z3 und Z4 und den Neumann'schen Rechnerkonzepten diskutieren zu können, stellen wir die Architektur der Zuse-Maschinen im Detail dar.

3 Konrad Zuses Maschinen Z1-Z4 und deren (geplante) Programmierung

Konrad Zuses Z1 (1938), Z3 (1941) und Z4 (1941-1945) verfügten über die folgende Architektur /ROJA98/:

- Freie Programmierbarkeit mit einem 8-Kanal-Lochstreifen (35 mm Film).
- Klare Trennung von Kontrolleinheit, Speichereinheit, arithmetischer Einheit, und den Ein- und Ausgabeeinheiten. Verbindung dieser Einheiten mit einem parallel arbeitenden Datenbus von 22 Bits.
- Parallel arbeitende Maschine mit 22 / 32 Bits (Z4 = 32 Bits) Datenübertragung.
- Implementierung von Mikrosequenzen in der Steuereinheit.
- Binäre arithmetische Einheit im Gleitkommasystem mit den vier Grundrechenarten und der Quadratwurzel (Z3). Die Z4 enthielt die zusätzlichen Operationen: x^2 , $\text{SQR}(x)$, $1/x$, $|x|$, $\text{sgn}(x)$, $x^{1/2}$, x^2 , $x \cdot (-1)$, x^{10} , x^3 , $x^{1/3}$, $x^{1/5}$, $x^{1/7}$, $x \cdot \pi$, $x^{1/\pi}$.
- Der einschrittige Übertrag bei der Addition- und Subtraktion.
- Verwendung der binären Gleitkommatechnik (ab 1936). Im Jahr 1985 /IEEE85/ hat IEEE im Standard 741 die Darstellung von Gleitkommazahlen standardisiert. Auffallend ist die fast identische Darstellung von Gleitkommazahlen im IEEE-Report und in der Zuse Z1 von 1936-1938.
- Umwandlung der binären Gleitkommazahlen in Dezimalzahlen und vice versa in den Ein- und Ausgabeeinheiten.
- Binäre Schaltungslogik (Schaltalgebra) in der gesamten Maschine (UND, ODER, NEGATION).
- Überlappende Ausführung der Instruktionen (pipelining) zur Geschwindigkeitssteigerung.

Wir illustrieren die Architektur der Z3 mit einem Blockschaltbild.

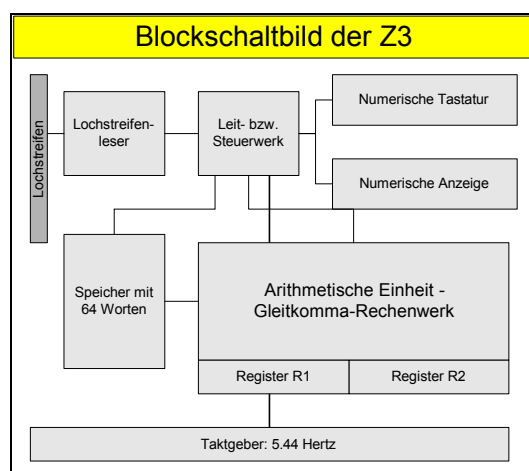


Bild 1: Architektur der Z3 (1941), die 1941 funktionsfähig war. Die Maschine Z3 verfügte über einen Lochstreifenleser (35 mm Normalffilm) zur Programmsteuerung, einem Steuerwerk zur Interpretation der Befehle und Steuerung der gesamten Maschine, einem Speicher mit 64 Worten a 22 Bits, einer

arithmetischen Einheit mit den Registern R1 und R2 à 22 Bits, den Ein- / Ausgabeeinheiten und einem Taktgeber von 5.4 Hertz. Der Datenbus arbeitete parallel (22 Bits). Die arithmetische Einheit konnte Gleitkommazahlen von 22 Bits Wortlänge (1 Bit Vorzeichen, 7 Bits Exponent, 14 Bits Mantisse) verarbeiten. Das Register R1 ist ein Schreib- und Lesespeicher à 22 Bits, während Register R2 nur ein Lesespeicher ist. Bei einer dyadischen arithmetischen Operation, z.B. der Addition, gilt: $R1 = R1 + R2$. Der Inhalt der Register R1 und R2 wird addiert und das Ergebnis wird Register R1. Das Register R2 wird auf leer gesetzt.

Das obige Blockschaltbild zeigt, daß die Z3 schon 1941 ein Computer war, wenn man der Definition von Ceruzzi folgt: *After 1945 a machine capable of the four operations of arithmetic, automatic storage and retrieval of intermediate results, and automatic input and output, all directed by a control unit.* Die Maschinen Z1- Z4 wurden durch ein starres Programm (Zuse nannte es starrer Rechenplan), welches nicht modifiziert werden konnte, gesteuert. Die Daten im Speicher konnten beliebig adressiert und modifiziert werden. Das Programm war nicht im Speicher der Maschine zusammen mit den Daten abgelegt, sondern auf 35mm Normalfilm gespeichert. Die Maschine Z3 wurde durch neun Instruktionen gesteuert.

Heutzutage wird der moderne Computer als John von Neumann Rechner bezeichnet. Wir präsentieren hier das Blockschaltbild des Neumann'schen Rechners.

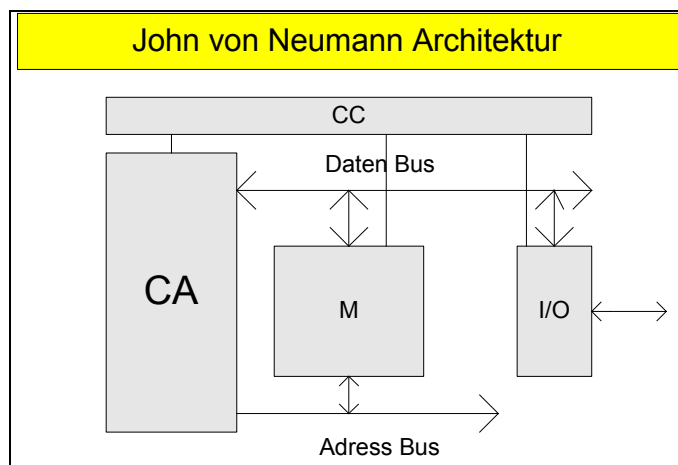


Bild 2: Der Neumann'sche Rechnertyp. Die Abkürzungen bedeuten /NEUM45/: CC: Logical control of the device, CA: Central arithmetic part, M: Total memory, I: Input, und O: Output.

Der Neumann'sche Rechner wird auch als *Stored-Program-Computer* (speicherprogrammierter Rechner) bezeichnet und besteht aus einem Speicher (M), einer arithmetischen Einheit (CA), einem Steuerwerk (CC), und den Ein- und Ausgabeeinheiten (I/O). Die Aufteilung des Speichers in einen ROM (read-only memory) und RAM (random-access-memory) haben wir weggelassen, da diese Unterscheidung hier unerheblich ist. Der ROM-Speicher wurde zur Speicherung von Konstanten vorgesehen. Das Programm befindet sich im RAM-Speicher und kann dort auch vom Steuerwerk (CC) durch das Programm selbst modifiziert werden.

Vergleichen wir die Blockschaltbilder der Z3 (funktionsfähig 1941) und des Neumann'schen Rechners (entworfen um 1945, aber nicht gebaut), dann fällt die starke Ähnlichkeit der beiden Maschinen auf. Beide Architekturen kennen eine Kontrolleinheit zur Interpretation der Befehle (freie Programmierbarkeit), den adressierbaren Speicher, die arithmetische Einheit und die Ein- / Ausgabeeinheiten. Der von Neumann Rechner speichert das Programm

zusammen mit den Daten im Speicher und das Programm kann dort auch durch das Programm selbst modifiziert werden. Die Speicherung des Programms mit den Daten war bei der Z3 nicht möglich. Rojas /ROJA98/, Seite 18, untersuchte, welcher Zusätze es in der Z3 bedurft hätte, um den Neumann-Rechner zu erhalten: *Automatische Adressensubstitution (indirect addressing) hätte im wesentlichen gereicht*. Der Neumann'sche Rechner ist somit **kein** neuer Rechnertyp, sondern es ist eine Erweiterung der Architektur der Zuse-Maschinen. Babbage kannte zwar auch die Trennung von zwei Komponenten eines Rechners (*store* und *mill*), aber seine Maschinen basierten aber nicht auf dem binären Zahlensystem und Schaltungsprinzip, welches sich heute bei modernen Computern fast ausnahmslos durchgesetzt hat.

Betrachten wir die Speicherung vom Programm im Speicher zusammen mit den Daten und der Modifizierbarkeit des Programms im Speicher durch das Programm selbst, dann finden sich weit vor den Ausführungen von John von Neumann /NEUM45/ und Burks et al. /BURK46/ deutliche Hinweise und Ideen in diese Richtung. Es darf auch nicht vergessen werden, daß Speicherplatz um 1945 und davor sehr teuer und platzraubend war, so daß die Verwendung des Speichers dazu nicht in Frage kam. Randell, ein international anerkannter Historiker auf dem Gebiet der Computerentwicklung, schreibt dazu /RAND73/, Seite 350: *The introduction of the Stored Program Electronic Computers: In fact the idea of a stored program has little attraction when a machine has only a slow mechanical internal memory. Thus it is not surprising that Zuse, whose 1936 /ZUSE36/ paper, reprinted earlier, contains a comparatively clear reference to the possibility of a stored program, does not seem to have persuaded the idea any further.*

Die Ausführungen von Randell sind gewichtige Argumente, die Speicherung des Programms und dessen Modifizierbarkeit außer acht zu lassen, wie das Atanasoff, Aiken, Eckert und Mauchly und Zuse aus guten Gründen taten. Konrad Zuse wollte umfangreiche Rechnungen mit seinen Maschinen ausführen, die Tausende von Instruktionen beinhalten sollten. Zuse schreibt dazu /ZUSE36/, Seite 1: *Vorliegende Erfindung dient dem Zweck, häufig wiederkehrende Rechnungen beliebiger Länge und beliebigen Aufbaus, die sich aus elementaren Rechenoperationen zusammensetzen, mit Hilfe von Rechenmaschinen selbsttätig durchzuführen*. Diese Rechnungen sollten nicht durch einen begrenzten Speicher beschränkt werden. Schon die Berechnung einer einfachen Determinante dritten Grades hätte den Speicher (Z3 und Z4) von 64 Worten an die Grenze seiner Kapazität /ZUSE98/ beansprucht. Konrad Zuse /ZUSE37/, Seite 2, linke Spalte, hat sich allerdings schon 1937 in einer Patentanmeldung klar zur Speicherung von Befehlen geäußert: *Bei der vorliegenden Erfindung werden diese Schaltglieder zu einem Speicherwerk zusammengesetzt, mit dem beliebige Angaben³, z.B. Zahlen, gespeichert werden können. Solche Vorrichtungen spielen besonders bei Rechenmaschinen eine Rolle. Sie können aber auch zur Speicherung anderer Angaben, z.B. von Befehlen an Arbeitsmaschinen (Schalt-speicherung), von Buchstabenkombinationen (z.B. Telegrammspeicherung), von Buchstabenverschlüsselungen (Chiffriermaschinen) od. dgl. benutzt werden.*

³ Konrad Zuse verwendete des öfteren den Begriff: *Angaben*. In einem Bericht von 1937 /ZUSE37a/ beschreibt Konrad Zuse sehr detailliert, was er unter Angaben versteht: *Diese können sehr verschiedene Bedeutungen haben, z.B. Zahlen, Aussagen, Namen, Kennziffern, Dienstgrade, Daten, Befehle, Nachrichten, Schlußfolgerungen, usw.* Er schreibt dann weiter: *Gemeinsam ist allen die Variabilität ihrer Aussage, denn wenn jede Ausgangsangabe nur eine Möglichkeit zuließe, so wäre das Rechnen überflüssig, da ja dann nur noch ein Resultat in Frage käme. So ist z.B. das Vorzeichen einer Zahl zweifach variabel, eine Dezimalzahl 10-fach, ein Buchstabe 26-fach, die Angabe über die Batallionszugehörigkeit dreifach variabel.*

Obige Ausführungen von Zuse beinhalten nur die Möglichkeit der Speicherung von Daten und Programmen zusammen, aber nicht der Modifikation der Programme durch das Programm selbst. In /ZUSE36/, Seite 6 unten, finden wir: *Auch der Rechenplan läßt sich speichern, wobei die Befehle in Takt der Rechnung den Steuervorrichtungen zugeführt werden. Die Rechenpläne lassen sich entsprechend in fester Form speichern, falls die Maschine oft dieselbe Rechnung ausführen soll.*

Wir sehen hier, daß Konrad Zuse sich durchaus vorstellen konnte, Befehle und Programme im Speicher abzulegen. Um die Ideen der Programmierung von Rechnern durch Zuse besser zu verstehen, zitieren wir für uns wichtige Passagen aus seinen Arbeiten zwischen 1936 und 1945.

In /ZUSE36/ finden wir: *Voraussetzung für jede Art der auszuführenden Rechnung ist die Aufstellung eines Rechenplans, in dem die aufeinanderfolgenden Rechenoperationen dem Charakter und der Reihe nach aufgezeichnet werden, und die im Verlauf der Rechnung auftretenden Zahlen fortlaufend numeriert oder nach einem anderen Schema geordnet werden, ohne sie zunächst der Größe nach zu bestimmen. Man geht von bestimmten Ausgangswerten aus, die den Variablen einer Formel entsprechen, und leitet aus diesen durch bestimmte Operationen über eine Reihe von Zwischenwerten die Resultatwerte ab. Ist für eine bestimmte Aufgabe ein solcher Rechenplan einmal aufgestellt, so gilt er für sämtliche Variationen der Ausgangswerte.*

Die Durchführung der zahlenmäßigen Rechnungen ist eine rein mechanische Tätigkeit. Sie läßt sich von Rechenmaschinen nach folgendem Verfahren durchführen.

Man verbindet die Rechenvorrichtung über ein Wählwerk mit einem Speicherwerk, das je Zelle eine Zahl aufnehmen kann. Das Wählwerk hat den Zweck, die erforderliche Speicherzelle mit der Rechenvorrichtung zu verbinden. sei es auf elektrischen oder mechanischen Wege, um entweder die gespeicherte Zahl zu einer Rechenoperation zu verwenden, oder um in der Zelle eine Zahl zu speichern. Das Speicherwerk dient zur Aufnahme der Ausgangswerte und der im Verlauf der Rechnung auftretenden Zahlen.

Auch der Rechenplan läßt sich speichern, wobei die Befehle in Takt der Rechnung den Steuervorrichtungen zugeführt werden. Die Rechenpläne lassen sich entsprechend in fester Form speichern, falls die Maschine oft dieselbe Rechnung ausführen soll. Alle diese Abwandlungen fallen unter das Grundprinzip.

Auf folgende Erweiterungen des Grundprinzips soll näher eingegangen werden: Bei vielen Rechnungen treten Werte auf, die konstant sind und sich nicht mit der Variation der Ausgangswerte ändern. Es wäre unpraktisch, diese für jede Rechnung mit den Ausgangswerten zusammen der Maschine zuzuführen. Sie gehören ihren Charakter nach zum Rechenplan, und es ist vorteilhaft, sie in den Lochstreifen des Rechenplans an den erforderlichen Stellen einzufügen. Der Abtaster muß in der Lage sein, die Angaben des Rechenplans sowohl als Befehl zur Steuerung als auch als Zahlen zur Verrechnung an die Maschine zu geben. Zu diesem Zweck wird auf dem Lochstreifen ein Feld freigehalten, das anzeigt, ob die Angabe des entsprechenden Abschnittes eine Zahl oder ein Befehl ist, und durch dieses Zeichen wird die vorübergehende Umstellung des Abtasters auf Zahlenangabe bewirkt. Die Angabe befindet sich vorzugsweise bereits bei dem der Zahl vorangehenden Steuerungsbefehl, damit die Umstellung schon während des Weiterrückens des Lochstreifens erfolgen kann.

Wichtig ist u.a. die Aussage: *Auch der Rechenplan läßt sich speichern, wobei die Befehle im Takt der Rechnung den Steuervorrichtungen zugeführt werden.* Es ist klar, daß wenn Rechenpläne gespeichert werden, dann auch die Adressen der Daten und evtl. auch in den Befehlen umgerechnet werden müssen.

In /ZUSE39/, Seite 2, können wir lesen: *Enthält der Rechenplan immer wiederkehrende Formeln, so können diese als Unterpläne ebenfalls gespeichert werden, so daß sie nicht immer neu angeführt zu werden brauchen. Dadurch geht die Aufstellung von Rechenplänen sehr schnell vonstatten. Kreis- und Hyperbelfunktionen usw. können als Potenzreihen gerechnet werden. Die entsprechenden Rechenpläne lassen sich in die Maschine fest einbauen und als Unterpläne verwenden.*

Hier gilt das schon oben gesagte, denn für die Abarbeitung der Unterpläne im Speicher dürfte eine Adressumrechnung der Daten notwendig sein.

Im Jahr 1938 /ZUSE38/ verläßt Zuse die starren Rechenpläne, die auf algebraischen Rechengeräten – z.B. Z4 - eingesetzt werden, und spricht von lebenden Rechenplänen. Im Jahr 1943 führt Zuse die logistischen Rechengeräte mit den freien Rechenplänen ein, die er 1938 noch lebendende Rechenpläne nannte. In /ZUSE43/ schreibt Zuse in der Zusammenfassung: *Mit Rechnen wird in diesem Bericht ganz allgemein das Rechnen mit Umständen und Bedingungen bezeichnet. Das Rechnen mit Zahlen ist nur ein spezielle Art dieses allgemeinen Rechnens, das durch den folgenden Satz definiert wird. Rechnen ist die Ableitung von Resultatangaben aus irgendwelchen Angaben nach einer Vorschrift.*

Zuse schreibt auf Seite 5: *Die algebraischen Rechengeräte dienen im wesentlichen den Aufgaben der reinen Zahlenrechnung, wobei die Rechenpläne starrer Natur sind, (d.h. ist der Rechenplan, nach dem gerechnet werden soll, einmal ausgewählt, so läuft er starr ab, wobei lediglich die in ihm auftretenden Zahlenwerte variabel sind). Für jede andere Rechnung muß erst ein anderer Plan gefertigt werden.*

Zuse führt jetzt (1943), Seite 6, die logistischen Rechengeräte ein: *Unter Logistik versteht man bekanntlich die mathematische Logik oder die Algebra der Logik, welche insbesondere die Aufgabe hat, die Richtigkeit mathematischer Sätze aus gegebenen Axiomen abzuleiten. Die Ausgangswerte der logistischen Rechnungen bestehen aus Umständen und Bedingungen, mit diesen wird gerechnet und das Resultat ermittelt. Unter Rechnen ist hier also nicht nur das Rechnen mit Zahlen, sondern ganz allgemein das Rechnen mit Umständen und Bedingungen zu verstehen.*

Auf Seite 8 führt Zuse die freien Rechenpläne ein: *Diese Rechenpläne heißen freie Rechenpläne im Gegensatz zu den starren Rechenplänen, weil bei Ihnen der Rechenablauf nicht nach einem starren Schema erfolgt, sondern weil sie je nach der gegebenen Aufgabe verschieden ablaufen. Ist solch ein freier Rechenplan einmal aufgestellt, so gilt er für alle Aufgaben eines bestimmten Typs, d.h. der Rechenplan gibt Antworten auf alle Fragen, die innerhalb eines fest umgrenzten Systems sinnvoll und lösbar sind. Bei den algebraischen Rechengeräten muss für jede dieser Aufgaben ein eigener Rechenplan aufgestellt werden, was natürlich erst bei umfangreichen Formeln wirtschaftlich ist. Bei den logistischen Rechengeräten ist nur ein einziger freier Rechenplan erforderlich.*

Auf Blatt 12 faßt Zuse die freien Rechenpläne zusammen: *Freie Rechenpläne: Selbsttätige Lösung verschiedenster Aufgaben innerhalb geschlossener Systeme. Kombinatorisches Rechnen mit Bedingungen, Umständen, Fällen usw. Selbsttätige Ausarbeitung von Rechenplänen.*

Konrad Zuse verwendet hier nicht das Wort speicherprogrammierter Rechner, um seine freien Rechenpläne realisieren zu können. Er schreibt vom kombinatorischen Rechnen mit Bedingungen, Umständen, Fällen usw. Dies führt zu den freien Rechenplänen. Aber um die freien Rechenpläne realisieren zu können, bedarf es nicht viel Phantasie, wie dies Konrad Zuse realisiert hätte, wenn der Ressourcenmangel im und nach dem zweiten Weltkrieg ihn daran nicht gehindert hätten. Auch Bauer /BAUE98/ diskutiert die Erfindung des

Neumann'schen Rechners und anerkennt bei Zuse: *Seine erst Jahre später publizierten visionären Ideen (Lebendige Rechenpläne) zielten in die richtige praktische Richtung, wurden von ihm aber nie verwirklicht.*

Zuse hatte die lebenden Rechenpläne /ZUSE38/, Seite 9, in einer Tagebuchnotiz schon 1938 erwähnt und diese dann später (1943) als freie Rechenpläne bezeichnet. Zuse schreibt in der Tagebuchnotiz von 1938 u.a.: *Abhängige und unabhängige Rückkopplung. Unabhängig = unabhängig von den Ausgangsangaben. Dient nur dem Zweck, den Plan konzentrierter darzustellen und ihn sich erst während der Arbeit entfalten zu lassen. Plan mit unabhängiger Rückkopplung ist noch als starr zu bezeichnen. Abhängig = Rückkopplung = eigentlich lebende Pläne. Einfluß der errechneten Angaben, also auch der Ausgangsangaben auf den Ablauf der Rechnung (Pläne). Zusammenfassung von Angaben zu Gruppen mit nur einer gemeinsamen Bezeichnung (z.B. Zahlen, Buchstaben, usw.).*

Entscheidend ist hier die Aussage: *Einfluß der errechneten Angaben, also auch der Ausgangsangaben auf den Ablauf der Rechnung (Pläne).* Bauer nennt kein Datum für Zuses erste Erwähnung der freien (lebendigen) Rechenpläne (Es war 1938). Es muß auch erwähnt werden, daß Zuse nach dem Krieg technisch und finanziell gar nicht die Möglichkeit hatte, die freien Rechenpläne in einer Maschine zu verwirklichen. Sogar die Amerikaner brauchten mit der EDSAC bis 1952, um den speicherprogrammierten Rechnertyp zu bauen. Konrad Zuse hat erst 1957 mit der Z22 den stored-program-computer verwirklichte.

Im Plankalkül /ZUSE45/, den Konrad Zuse von 1942-1945/46 (Endfassung) erstellte, finden sich u.a. die Zuweisung von Variablen, Datenstrukturen, Schleifen, und Operationen des Prädikatenkalküls. Im Plankalkül - erstmals veröffentlicht 1972 /ZUSE72/ - hat Zuse u.a. die Verbindung zwischen der Prädikatenlogik, der Arithmetik und der Struktur (Bitstruktur) einer Rechenmaschine hergestellt. Die Idee der freien Rechenpläne auf abstraktem Niveau (nicht implementiert) ist hier verwirklicht. Konrad Zuse hat mit dem Plankalkül u.a. das Schachspiel demonstriert. Auch hier finden sich Ideen, die nur durch die Modifikation von Programmen und deren Ablauf im Speicher realisiert werden können.

Wir wollen nun die Ausführungen von John von Neumann im Detail betrachten.

4 Der John von Neumann Rechner

Wir wollen die postulierte Architektur eines Computers durch John von Neumann an Hand der Artikel von Neumann /NEUM45/, Burks et al. /BURK46/ und Mauchly /MAUC47/ diskutieren.

4.1 Der First Draft of a Report on the EDVAC (June 30, 1945)

Von Neumann /NEUM45/ definiert ein automatisches Computersystem: 1.2 An automatic computing system is a (usually highly composite) device, which can carry out instructions to perform calculations of a considerable order of complexity - e.g. to solve a non-linear partial differential equation in 2 or 3 independent variables numerically.

Charles Babbage und Konrad Zuse haben die freie Programmierbarkeit eingeführt, wobei Konrad Zuses Rechner Z1 und Z3 in der Speicheradressierung und mit der binären arithmetischen Einheit flexibler als die Babbage Rechner waren. Zuses Rechner waren frei programmierbare Rechner, d.h. sie konnten eine Folge von sinnvollen Instruktionen verarbeiten und entsprechende Speicheradressierungen und arithmetische Gleitkommaoperationen ausführen.

Interessant ist die Anmerkung von Neumanns in 1.4: *The remarks of 1.2 on the desired automatic functioning of the device must, of course, assume that its functions are faultlessly. Malfunctioning of any device has, however, always a finite probability - and for a complicated device and a long sequence of operations it may not be possible to keep this probability negligible. Any error may vitiate the entire output of the device. For recognition and correction of such malfunctions intelligent human intervention will in general be necessary.*

Es soll hier erwähnt werden, daß Zuses Z3 und Z4 über eine arithmetische Ausnahmebehandlung verfügten, die Rechenfehler vermied.

In 2.1, 2.2 beschreibt von Neumann die elementaren Operationen der Arithmetik, über die eine solche Maschine verfügen soll. Neben den Grundoperationen +, -, *, /, sind auch Wurzeloperationen und Logarithmen aufgeführt. Konrad Zuse hat in seinen Maschinen Z3 (1941) und Z4 (1942-1945) solche Operationen implementiert, und zwar in binärer Gleitkommatechnik (siehe oben). Logarithmische Funktionen wollte er durch Unterpläne realisieren, wie z.B. die hyperbolischen Funktionen (siehe oben).

In 2.3 führt Neumann die *logical control of device* ein, welche nichts anderes macht, als die Instruktionen an die Maschine zu interpretieren und die Maschine entsprechend zu steuern. Solch eine Kontrolleinheit hatte Konrad Zuse bereits mit der Z1 (1938) implementiert.

In 2.4 diskutiert von Neumann den Speicher einer Maschine und faßt dessen Funktion in 2.5 zusammen: *The device requires a considerable memory. While it appeared, that various parts of the memory have to perform functions which differ somewhat in their nature and considerably in their purpose, it is nevertheless tempting to treat the entire memory as one organ, and to have its parts even as interchangeable as possible for the various functions enumerated above.*

Die Aufgaben des Speichers sollen die Speicherung von Zwischenwerten von arithmetischen Operationen sein, die Bereithaltung von Instruktionen, Funktionen und Randbedingungen für Differentialgleichungen. In /RAND73/, Seite 350, 3. Absatz, schreibt Randell: *It is generally accepted that the first documented discussion of these topics, and the advantages of using just one large internal memory, in which instructions as well as data be held, was the draft report of the EDVAC written by von Neumann.* Aus unserer Sicht schreibt von Neumann hier nicht explizit, daß Daten und Instruktionen zusammen gespeichert werden sollen, auch aus 2.4 a-h geht dies nicht eindeutig hervor. Er spricht zwar von *as one organ*, aber dies ist uns ein wenig vage.

Von Neumann führt dann noch *das outside recording medium R* ein und fordert in 2.8. die Möglichkeit der Darstellung der Ergebnisse auf einer Ausgabeeinheit. Damit hat von Neumann fünf Komponenten (siehe auch Bild 2) für einen Computer definiert:

1. Die arithmetische Einheit (CA=Central arithmetical part)
2. Kontrolleinheit (CC=Logical control of the device)
3. Speicher (M=Total memory)
4. Eingabeeinheit (I= Input)
5. Ausgabeeinheit (O= Output).

Auf Seite 358, Mitte, finden wir: *Could not all, or at least some functions of M – preferably those which involve great bulks of information – be taken over by R?* Dies ist eine interessante Anmerkung, da hier das Speicherband der Turingmaschine gemeint sein könnte. Weiter unten

schlägt von Neumann für R einen *stack of punchcards* oder *a length of teletype tape* vor. In Kapitel 4 diskutiert Neumann im wesentlichen die Zustände in der Natur und kommt zu dem Schluß, daß auch Neuronen den *all-or-none character, that is two states*, haben. Er schreibt auf Seite 361, 1. Absatz: *It is easily seen, that these simplified neuron functions can be imitated by telegraph relays or by vacuum tubes*

Bis hierher muß festgestellt werden, daß die Ausführungen von John von Neumann in den Kapiteln 1-4 eine Rechnerarchitektur beschreibt, die Zuse in seinen Maschinen Z1 (1938, keine arithmetische Ausnahmebehandlung), Z3 (1941) und Z4 (1942-1945) im wesentlichen bereits realisiert und implementiert hatte. In der Z4 hatte er mit sechs Lochstreifenlesern eine ausgefeilte Unterprogrammtechnik implementieren wollen, mit der er immer wiederkehrende Rechenpläne abarbeiten wollte. Nur der Materialmangel im Krieg verhinderte eine Implementierung. In /ZUSE36/, Seite 6, schreibt Konrad Zuse: *Es lassen sich mehrere Rechenwerke, Speicherwerke, Verteiler, Abtaster, Locher usw. einbauen und somit mehre Operationen zugleich ausführen. In festen Zahlenspeichern können häufig gebrauchte Zahlen, wie $\sqrt{2}$, π , g ständig zur Verfügung gehalten werden.* Dies ist der ROM-Speicher, wie von Neumann definiert, aber Zuse geht weiter, da er von mehreren Rechenwerken spricht. Leistungsfähige Ein- und Ausgabeeinheiten mit binärer Gleitkomma-Dezimal Umwandlung waren schon in der Z1 und Z3 implementiert.

Betrachten wir nun das Kapitel 5. Von Neumann schlägt eine Realisierung seiner Maschine mit Röhren vor und schreibt: *This suggest the use of the binary system. .. A consistent use of the binary system is also likely to simplify the operations of multiplications and divisions considerably... Hence M should store binary material only. This necessitates incorporating decimal-binary and binary-decimal conversion facilities into I and O.* Auf den folgenden Seiten diskutiert Neumann das Problem der binären Multiplikation und die Ein- und Ausgabeeinheiten mit der Konvertierung von binären Zahlen in das Dezimalsystem und vice versa. Solche Ein- und Ausgabeeinheiten waren bereits in der Z1 (1938) implementiert und in der Z3 (1941) voll funktionsfähig.

In Kapitel 5 diskutiert von Neumann den Aufwand für verschiedene arithmetische Operationen Auch hier findet sich nichts Neues, was nicht schon durch Konrad Zuse in seinen Maschinen Z1-Z4 implementiert worden wäre. Konrad Zuse hat nie etwas anderes als das binäre Prinzip betrachtet und bei der Realisierung der arithmetischen Einheit war es klar, daß arithmetische Operationen für große und kleine Zahlen möglich sein mußten, was von Neumann nicht diskutiert wird.

4.2 Der Artikel von Burks et al.: Preliminary Discussion of the Logical Design of an Electronic Computing Instrument

Wir diskutieren jetzt das Papier von Burks et al. /BURK46/, wo u.a. die Speicherung des Programms zusammen mit den Daten vorgeschlagen wird. Auf Seite 371 lesen wir: *Inasmuch as the completed device will be a general-purpose computing machine it should contain main organs relating to arithmetic, memory-storage, control and connection with the human operator. It is intended that the machine be fully automatic in character, i.e. independent of the human operator after the computation starts.*

Dieses Konzept ist bereits zwischen 1938 und 1941 in den Maschinen Z1 und Z3 realisiert worden. Besonders interessant ist der Begriff *human operator*. Die Maschine Z3 verfügte über einen Betriebsmodus, der es erlaubte, mit der Maschine zu kommunizieren, d.h. ein

Programm konnte gestoppt werden, um sich die Zwischenergebnisse anzuschauen und neue Dateneingaben und Zwischenrechnungen vorzunehmen. Es war aber auch möglich, die Maschine Z3 im Sinn von *independent of the human operator after the computation starts* arbeiten zu lassen, da z.B. die arithmetische Ausnahmebehandlung garantierte, daß die Maschine korrekt rechnete. Die Z3 konnte auch mit den Werten plus / minus-unendlich korrekt rechnen.

Burks et al. schreiben dann weiter:

It is evident that the machine must be capable of storing in some manner not only the digital information needed in a given computation and also the intermediate results of the computation, but also the instructions which govern the actual routine to be performed on the numerical data. In a special-purpose machine these instructions are an integral part of the device and constitute a part of its design structure. For an all-purpose machine it must be possible to instruct the device to carry out any computation that can be formulated in numerical terms. Hence, there must be some organ capable of storing these program orders. There must, moreover, be a unit which can understand the instructions and order their execution.

Consequently we have discussed above two different forms of memory: storage of numbers and storage of orders. If, however, the orders to the machine are reduced to a numerical code and if the machine can in some fashion distinguish a number from an order, the memory organ can be used to store both numbers and orders.

Diese beiden oft zitierten Absätze beinhalten, daß neben den Daten auch die Befehle im Speicher abgelegt werden sollen. Ob Burks et al. mit *and order their execution* eine Modifikation von Befehlen meinten, ist für uns sehr vage. Dies ist aber kein neuer Rechnertyp, denn es ist unerheblich, ob die Befehle zusammen mit den Daten gespeichert werden, oder die Befehle sich auf einem extra Speichermedium befinden. Entscheidend ist, ob die Befehle modifiziert werden sollen, z.B. durch eine Adressumrechnung. Dies ist aus den obigen Äußerungen aber nicht klar zu erkennen. Erst Mauchly (siehe unten) wird hier präzise.

If the memory for orders is merely a storage organ there must be an arithmetic organ which can automatically execute the orders by the memory. We shall call this organ the control.

Inasmuch as the device is a computing machine there must be an arithmetic organ in it which can perform certain of the elementary arithmetic operations. There will be, therefore, a unit capable of adding, subtracting, multiplying and dividing.

The operations that the machine will view as elementary are clearly those which are wired into the machine. To illustrate, the operation of multiplication could be eliminated from the device as an elementary process if one were willing as a properly ordered series of additions. Similar remarks apply to division. In general, the inner economy of the arithmetic unit is determined by a compromise between the desire for speed of operation - a non-elementary operation will generally take a long time to perform since it is constituted of a series of orders given by a control - and the desire for simplicity, or cheapness, of the machine.

Lastly, there must exist devices, the input and output organ, whereby the human operator and the machine can communicate with each other.

Dies beschreibt einerseits das Konzept der Trennung der Operationseinheiten in der Maschine, aber auch die Realisierung der arithmetischen Operationen. Zuse hatte in der Z3 beides realisiert und z.B. die Multiplikation auf wiederholte Additionen zurückgeführt /ROJA98/, Seite 45ff. Dazu verwendete er eine Mikrosequenzsteuerung mit Schrittschaltern.

In Kapitel 2 diskutieren Burks et al. den Speicher. Hier ist folgender Satz interessant: *Not only must the memory have sufficient room to store these intermediate data, but there must be provisions whereby these data later can be removed, i.e. at the end of the $(t + dt)$ cycle, and replaced by the corresponding data from the memory and of replacing them with new information must, of course, be done quite automatically under the direction of the control.*

Dieses Prinzip war bereits in der Z1 (1938) implementiert. Dort hatte Zuse die Register R1 und R2 eingeführt und eine Addition erfolgte nach folgendem Prinzip: $R1 = R1 + R2$. Der Inhalt von Register R2 konnte durch neue Daten aus dem Speicher ersetzt werden.

Im Kapitel 3 diskutieren Burks et al. Operationen im Speicher mit Daten und Instruktionen. Wichtig erscheint uns folgendes: *To summarize, transfers into the memory will be of two sorts: Total substitutions whereby the quantity previously stored is cleared out and replaced by a new number. Partial substitutions in which that part of an order containing a memory location number – we assume the various positions in the memory are enumerated serially by the memory location-numbers – is replaced by a new memory location number.*

Dieses Prinzip war in den Maschinen Z3 und Z4 **nicht** implementiert. Die obigen Anmerkungen von Burks et al. beziehen sich zwar auf die Speicherung von Befehlen im Speicher und der Modifikation von Speicheradressen. Es ist hier noch nicht die Modifikation von Befehlen zu erkennen, sondern eben nur die Modifikation von Speicheradressen. Es ist aber in diesem Artikel der einzige Hinweis auf die Speicherung von Befehlen im Speicher mit den Daten und der Modifikation von Adressen.

In Kapitel 4 werden der Speicher und dessen technischer Aufbau diskutiert, auch wird eine Hierarchie von Speichern diskutiert. Zuse verfolgte ebenfalls eine Hierarchie von Speichern, indem er überlegte, häufig benötigte Unterpläne mit den Zahlen (Konstanten) im Speicher abzulegen.

In Kapitel 5 wird die arithmetische Einheit diskutiert. Wichtig erscheint uns hier: *It would therefore seem to us not at all clear whether the modest advantages of a floating binary point offset the loss of memory capacity and the increased complexity of the arithmetic and control circuits.*

Hier sehen Burks et al. noch Probleme bei den Gleitkommazahlen. Burks et al. schreiben hier u.a., daß die Wurzelrechnung durch iterative Techniken (Division) realisiert werden können. Dies war auch schon in der Z3 realisiert /ROJA98/.

Kapitel 6 ist wieder eine Diskussion von technischen Details der Kontrolleinheit.

4.3 Article of Mauchly: Preparation of Problems for EDVAC-Type Machines

Erst im Artikel von Mauchly /MAUC47/, Seite 365, Mitte, finden sich konkrete Aussagen zur Modifikation von Instruktionen im Speicher: *(3) ability to store instructions as well as numerical quantities in the internal memory, and modify instructions so stored in accordance with other instructions.*

Auf Seite 366, Mitte, finden wir: *One can, therefore, modify not only the numbers which are substituted in a process, but the process itself, in any desired systematic way.*

The third characteristic is that the instructions are stored in the internal memory in the same manner as are numerical quantities, and one set of instructions can be used to modify another set of instructions.

Das letzte Zitat ist der deutlichste Hinweis auf die Modifikation von Instruktionen im Speicher durch die Instruktionen selbst.

5 Der Neumann'sche Rechner - Eine Bewertung der Prioritäten

Nach einer sorgfältigen Analyse der Artikel von Neumann /NEUM45/, Burks et al. /BURK46/ und Mauchly /MAUC47/, ist die Situation aus unserer Sicht folgende: Es ist **nicht** das Verdienst von John von Neumann, Burks et al. und Mauchly, die binäre Schaltungslogik, die Binärzahlen, die binären Gleitkommazahlen, die Auflösung arithmetischer Operationen in elementare Operationen, oder die Trennung der Operationseinheiten in einem Computer als erster postuliert zu haben. Die in den obigen Papieren zwischen 1945 und 1947 angeführten logischen Strukturen eines Rechners wurden von Konrad Zuses ab 1936 nicht nur diskutiert, sondern in der Z1 (1938) und Z3 (1941, funktionsfähig) sogar implementiert. Konrad Zuses arithmetische Einheit rechnete mit binären Gleitkommazahlen und verfügte über eine arithmetische Ausnahmebehandlung, die in den Papieren von Burks /BURK46/ und von Neumann /NEUM45/ nur angedacht werden, indem gesagt wird, daß korrekt gerechnet werden muß, bzw. keine Fehler (malfunctions) auftreten dürfen. Es ist auch **nicht** das Verdienst von John von Neumann und den anderen Autoren, Röhren als binäre Bauelemente für Rechner vorgeschlagen zu haben, letzteres hat Helmut Schreyer /GEBH83/, /SCHR41/ in seiner Dissertation schon 1941 beschrieben und zusammen mit Konrad Zuse weiterentwickelt. Auch der Rechner ABC von Atanasoff war mit 600 Röhren bestückt und rechnete im Binärprinzip, wenn auch ohne Komma und ohne Programmierung (Integerzahlen). Auch COLOSSUS arbeitete mit Röhren im Binärprinzip, war aber nicht frei programmierbar.

Auch andere Autoren teilen unsere kritische Sicht zum Neumann'schen Rechner. Wir zitieren Passagen aus Bauer /BAU98/, der ausführt, daß man Rechner als stored-program-computer bezeichnet, deren Befehle während des Programmablaufs abgeändert werden können. Er führt dann aus, daß man dies fast gratis erhält, wenn man Befehle und Daten im selben Speicher ablegt. Dann schreibt Bauer: *Eben diese Idee hatten schon im Januar 1944 die Chefs des Teams, das die ENIAC entwickelte, J. Presper Eckert und John Mauchly, die sich Gedanken über eine weit leistungsfähigere und weit flexiblere Nachfolgermaschine machten. Im weiteren Verlauf, jedenfalls vor dem 30. Juni 1945, hatte Eckert dann die Idee, in der künftigen EDVAC Befehle durch die Maschine abändern zu lassen.* Im letzten Absatz schreibt dort F.L. Bauer: *Trotzdem bleibt die historische Forschung heute dazu geneigt, in erster Linie Eckert und Mauchly, in zweiter Linie von Neumann, Goldstine und Burks die Erfindung des (zahn) speicherprogrammierten Rechners, das potentiellen Computers zuzubilligen.* Auch schreibt Bauer u.a.: *Historiker wie Paul Ceruzzi, Brian Randell ... vermeiden deshalb in der Regel den inkorrekten Ausdruck von "Neumann computer";.....sollte man in Deutschland "speicherprogrammierter Rechner" sagen, wenn ein solcher gemeint ist, und nicht "von-Neumann-Rechner".*

In /ROJA98/, Seite 17, schreibt Bauer, indem er Güntsch auf einem Vortrag in Weimar (GÜNT96) zitiert: *Adressenberechnungen eingeschlossen, ist Zuse hier offensichtlich (1936!) fast soweit wie Mauchly 1947 (...). Aber noch hat er nur starre Rechenpläne, weit entfernt von der Universalität der Turingmaschine. Am 4.-5. Juni 1938 notiert Zuse dann eigentlich lebende Rechenpläne... Später wird Zuse das freie Rechenpläne nennen. "Ist Zuse damit nun auch der Vater des von Neumann Rechners, fragt Güntsch und fährt fort: Zuse hat ohne Zweifel 1938 das Grundprinzip klar formuliert und seine Gedanken dazu schriftlich festgehalten".*

In Rojas /ROJA98/, Seite 18, oben, schreibt dann Bauer: *Es ist wohl auch erlaubt, zu spekulieren, was Zuse im Plankalkül gemacht hätte, wenn ihm die Aufgabe gestellt worden wäre, die Ackermannfunktion $ack(4,4)$ zu berechnen oder gar ein allgemeines Programm zu schreiben für die Ackermann-Funktion $ack(x,y)$ ⁴.*

Wir möchten noch ein paar Anmerkungen zur Universalität von Rechnern im Sinne der Turingmaschine machen. Eine Maschine mit ausreichendem Speicher, der sowohl Daten wie Befehle faßt, und mit einem zur Ausführung der Befehle CLR (löschen), INC (inkrementieren), LOAD (Lesen), STORE (speichern) und BR (springen falls Null) fähigen Prozessor ist eine universelle Turingmaschine (der Befehlssatz kann weiter reduziert werden /ROJA96/). Rojas /ROJA98/ hat 1998 bewiesen, daß auch Zuses Z3 eine universelle Maschine war. Wir halten die Diskussion, die frühen Maschinen daran zu messen, ob sie eine universelle Maschine waren oder nicht, nicht für sehr hilfreich für die praktische Anwendung der Maschinen. Eine Maschine mit den fünf oben angeführten Befehlen ist in der Praxis **nicht** einsetzbar, obwohl sie universell ist. Man stelle sich nur einmal vor, wie eine Gleitkommaaddition damit ausgeführt werden soll. Aus unserer Sicht ist wichtig, ob die anfallenden (mathematischen) Probleme von der Maschine effizient gelöst werden können. Ist dies der Fall, dann ist es eine brauchbare Maschine. Wird dann auch noch mathematisch bewiesen, daß diese Maschine universell im Sinne von Turing ist, ist es um so besser. Diese Sicht hat sich in der Praxis bestätigt, nämlich als z.B. die Z4 von 1950-1955 an der ETH-Zürich hunderte numerischer Probleme berechnete. Die Frage des rekursiven Aufrufs, wie z.B. bei der Ackermannfunktion, war damals kein Diskussionsgegenstand aus der Praxis.

Wie oben aufgezeigt wurde, geht die konkrete Formulierung der Modifikation von Programmen im Speicher durch das Programm selbst wohl eher auf Mauchly / Eckert als auf John von Neumann zurück. Mauchly hat es explizit ausgeführt. Allerdings müssen auch die Ausführungen von Konrad Zuse mit den lebenden Rechenplänen (1938), den freien Rechenplänen (1943) und der logistischen Maschinen beachtet werden, die mit anderen Worten als bei Mauchly eine sehr flexible Programmierung und damit auch Rechnerstrukturen beschreiben.

6 Zusammenfassung

Es ist unbestritten, daß John von Neumann ein hervorragender Wissenschaftler war und seine Arbeiten weltweite Anerkennung verdienen, aber seine Verdienste bei der Konstruktion von Computern und besonders deren logischer Struktur sind eher minimal. Die EDVAC (1952), war nicht der erste fertiggestellte stored-program-computer, es war die EDSAC (UK, Mai 1949). Die konkrete Formulierung der Modifikation von Programmen im Speicher durch das Programm selbst geht wohl eher auf Mauchly / Eckert als auf John von Neumann zurück. Allerdings müssen auch die Ausführungen von Konrad Zuse mit den lebenden Rechenplänen (1938), den freien Rechenplänen (1943), die Idee der logistischen Maschinen (1943) und des Plankalkül beachtet werden.

Logische Grundlagen der Computerentwicklung, wie arithmetische Einheiten basierend auf dem Binärsystem, oder die Trennung von Kontrolleinheit, Speicher, usw. sind die Verdienste von Konrad Zuse und teilweise auch von Babbage (*store* und *mill*). Konrad Zuse hat das

⁴ Dies ist eine interessante Anmerkung von Bauer. Zuse testete den Plankalkül mit Aufgaben aus den (Ingenieur)wissenschaften und die Programmierung des Schachspiels schien ihm die größte Herausforderung zum *Test* des Plankalküls zu sein. Der Plankalkül enthält keine Sprachkonstrukte für den rekursiven Aufruf von Programmen. Zuse wurde bei seinen vielen Programmbeispielen /ZUSE72/ anscheinend nicht mit dem Problem der Rekursivität konfrontiert.

Prinzip der Trennung von Kontrolleinheit, Speicher, arithmetischer Einheit und den Ein- / Ausgabeeinheiten schon 1938 mit der Z1 verwirklicht und mit der Z3 (1941) war es in einer funktionsfähigen Maschine implementiert. Das von den Autoren Neumann, Burks et al. und Mauchly propagierte Binärprinzip war ebenfalls schon in der Z1 (1938) von Zuse verwirklicht. Bauer /BAUE98/, aber auch Randell /RAND73/, Seite 351, 13. Zeile von unten, rechnen den *stored program computer* nicht John von Neumann zu, sondern eher Mauchly /MAUC47/. Bauer /BAUE98/ zitiert Güntsch /GÜNT96/, der argumentiert, daß Zuse schon 1938 /ZUSE38/ die Grundprinzipien des Neumann'schen Rechners erkannt und in Tagebuchnotizen niedergeschrieben hat.

Wie auch immer der Neumann'sche Rechnertyp in Zukunft historisch eingeordnet werden wird, ist es unser Anliegen, Beiträge zur Entwicklung des Computers zu liefern, und dies auch kontrovers zu diskutieren. Es ist unsere Überzeugung, daß noch umfangreiche Forschungsarbeiten - z.B. in Form von Dissertationen - notwendig sind, um die Entwicklung des Computers historisch korrekt darzustellen.

7 Literatur

Die unten angeführten Artikel von Konrad Zuse sind im Heinrich Nixdorf MuseumsForum in Paderborn oder bei Horst Zuse erhältlich. Raul Rojas von der Freien Universität Berlin erstellt z.Z. ein Internet-Archiv über Konrad Zuse mit Unterstützung der DFG unter der URL: <http://www.zib.de/zuse>.

/BAUE98/ Bauer, Friedrich, L.: Wer erfand den Neumann'schen Rechner? Informatik Spektrum, April 98, pp. 84ff.

/BURK46/ Burks, A.W.; Goldstine, H.H. Neumann, John von: Preliminary Discussion of the Logical Design of an Electronical Computing Instrument, 1946. In: Taub, A.H. (Editor), Collected Works of John von Neumann, Vol. 5, New York, Macmillan, 1963, pp. 34-79. Also in Randell von 1973 /RAND73/, Seiten 371ff.

/GEBH83/ Gebhardt, Friedrich (Editor): Skizzen aus den Anfängen der Datenverarbeitung, Berichte der Gesellschaft für Mathematik und Datenverarbeitung, Bericht 143. Oldenburg Verlag, 1983. Enthalten sind zwei Berichte von Helmut Schreyer.

/GOLD72/ Goldstine, Hermann, H.: The Computer from pascal to von Neumann.

/GÜNT96/ Güntsch, Fritz-Rudolph: Vortrag anlässlich des Konrad Zuse Symposiums, März 1996 in Weimar.

/IEEE85/ IEEE: IEEE Standard for Binary Floating Point Arithmetic (ANSI / IEEE Std 754-1985), New York, August 1985.

/LEGE83/ Legendi, T.; Szentivanji, T.: Leben und Werk von John von Neumann, BI-Wissenschaftsverlag, 1983

/MAUC47/ Mauchly, John, W.: Preparation of Problems for EDVAC-Type Machines. In /RAND73/, Seiten 365ff.

/NEUM45/ Neumann, John von: First Draft of a Report on the EDVAC. Moore School of Electrical Engineering, University of Pennsylvania, June 30, 1945. In /RAND73/, Seiten 355ff.

/RAND73/ Randell, B. (Editor): The Origins of Digital Computers. Springer Verlag, Heidelberg, New York, 1973.

/RAND73a/ Randell, B. (Editor): Stored Program Electronic Computers, Chapter VIII. In: The Origins of Digital Computers. Springer Verlag, Heidelberg, New York, 1973, pp. 349ff.

- /ROJA96/ Rojas, Raul: Conditional Branching is not Necessary for Universal Computation in von Neumann Computers. Journal of Universal Computer Science, Band 2, Nr. 11, 1996, S. 756ff.
- /ROJA97/ Rojas, Raul: Konrad Zuses Rechenmaschinen: Sechzig Jahre Computergeschichte.
- /ROJA98/ Rojas Raul (Editor): Die Rechenmaschinen von Konrad Zuse, Springer Verlag, 1998.
- /ROJA98a/ Rojas, Raul: How to make Zuse's Z3 a Universal Computer. IEEE Annals of Computing, Vol. 20, No. 3, July/Sept. 1998.
- /SCHR41/ Schreyer, Helmut: Schaltungsanordnung eines elektrischen Kombinationspeicherwerkes. Dissertation, GMD-Nr. 004/001, 1941, Patentschrift Nr. 937 170, ausgegeben am 29. Dezember 1955.
- /STAC98/ Heike Stach: Als Rechner zu abstrakten Maschinen wurde. In: Zur Sozialgeschichte der Informatik, 2 / 98.
- /ZUSE36/ Zuse, Konrad: Verfahren zur selbsttätigen Durchführung von Rechnungen mit Hilfe von Rechenmaschinen. Patentanmeldung Z 23 139 / GMD Nr. 005/021 / 11.4.1936.
- /ZUSE37/ Zuse, Konrad: Aus mechanischen Schaltgliedern aufgebautes Speicherwerk. Patentschrift Nr. 924 107, 3. Juli 1937.
- /ZUSE37a/ Zuse, Konrad: Einführung in die allgemeine Dyadik.
- /ZUSE38/ Zuse, Konrad: Tagebuchnotiz vom 4.6.38 zur Computerarchitektur.
- /ZUSE39/ Zuse, Konrad: Rechenmaschine. Bericht von Konrad Zuse als Gefreiter, 1939.
- /ZUSE39/ Zuse, Konrad: Selbsttätige Rechengeräte. GMD: 009 006, 1943.
- /ZUSE45/ Zuse, Konrad: Theorie der angewandten Logistik - 2. Buch. der Zuse Apparatebau Berlin, 1945. Siehe auch /ZUSE72/
- /ZUSE47/ Zuse, Konrad: Über die Mechanisierung Schematisch - Kombinativer Aufgaben. Zuse Ingenieurbüro Hopferau, 1947.
- /ZUSE49/ Zuse, Konrad: Rechenmaschine zur Durchführung von arithmetischen Rechenoperationen. Patentschrift: Nr. 975 966, 30. Eingereicht Juni 1949, Erteilt: 6. Dezember 1962.
- /ZUSE72/ Zuse, Konrad: Der Plankalkül. Gesellschaft für Mathematik und Datenverarbeitung. Nr. 63, BMBW - GMD - 63, 1972.
- /ZUSE86/ Zuse, Konrad: Der Computer - Mein Lebenswerk. 2. Auflage, Springer-Verlag, 1986.
- /ZUSE93/ Zuse, Konrad: The Computer – My Life. Springer Publisher, 1993.
- /ZUSE98/ Zuse, Horst: Konrad Zuse Multimedia Show. Hypertextsystem mit 330 Seiten, 500 weitgehend unbekannte Bilder, 30 weitgehend unbekannte Videos, 1998. Erhältlich bei: <http://home.t-online.de/home/horst.zuse>.

Horst Zuse

Dr.-Ing. Horst Zuse wurde am 17. November 1945 in Hindelang im Allgäu geboren. Er studierte von 1967 bis 1973 Elektrotechnik an der Technischen Universität Berlin (TUB). Seit 1975 ist er wissenschaftlicher Angestellter im Fachbereich Informatik der TU-Berlin.

Von 1975 bis 1982 war er maßgeblich an der Entwicklung des Information-Retrieval-Systems FAKYR (Fachbereich-Kybernetik-Retrievalsystem) beteiligt. Es diente zur Überprüfung von Hypothesen auf dem Gebiet des Information-Retrieval. Eine von Dr. Horst Zuse modifizierte Version des Systems FAKYR wird seit 1989 in der Fachbibliothek Informatik eingesetzt.

1985 promovierte er auf dem Gebiet der Softwarekomplexitätsmaße.

Im Mai 1987 folgte ein einjähriger Forschungsaufenthalt am *IBM Thomas J. Watson Research Center in Yorktown Heights / New York* in den USA. Dort arbeitete er im Projekt PUNDIT (Program-Understanding-Investigation-Tool). Aufgabe dieses Projektes war die Analyse von *Old-Code Software* mit Softwaremaßen. Die Ergebnisse der Arbeiten auf dem Gebiet der Softwaremessung wurden 1991 im Buch: *Horst Zuse: Software Complexity - Measures and Methods*, vom DeGruyter Verlag veröffentlicht.



Zurück an der TU-Berlin, setzte er die Forschungsarbeiten auf dem Gebiet der Softwarequalitätsmessung, insbesondere die Anwendung von Softwaremaßen in der Praxis, fort. Ein weiterer Schwerpunkt seiner Arbeiten war die Fundierung der Validation von Softwaremaßen und die Analyse von qualitativen Vorhersagemodellen für den Software-Lebenszyklus.

Von 1989 bis 1995 gab er mehrfach im Jahr 3-Tages Seminare auf dem Gebiet der Quantifizierung der Softwarequalität im Rahmen von DecCollege und ORACLE für Entwicklungsabteilungen der Industrie. Ziel dieser Seminare war u.a. die Einführung von Methoden der Softwarequalitätsmessung für die Entwicklung kommerzieller Software. Im selben Zeitraum war er zu vielen Präsentationen und Seminaren in Universitäten und auf Konferenzen im europäischen Ausland, den USA und Australien eingeladen.

Von August 1994 bis März 1995 war er als Gastforscher von der *Gesellschaft für Mathematik und Datenverarbeitung (GMD)* im Institut für *Anwendungsorientierte Software- und Systemtechnik* eingeladen.

Im Dezember 1997 erschien das Buch: *Horst Zuse: A Framework of Software Measurement*, DeGruyter Verlag. Dieses Buch präsentiert Konzepte, Methoden und Werkzeuge auf dem Gebiet der quantitativen Methoden des Softwareengineering. Auf 750 Seiten wird dargestellt, wie Softwaremaße theoretisch zu verstehen sind, und wie sie in der Praxis eingesetzt werden können. Parallel zum Buch wurde von 1993-1997 die Entwicklung des Systems ZD-MIS zusammen mit Karin Drabe durchgeführt. ZD-MIS enthält u.a. eine Datenbank von mehr als 1500 Softwaremaßen, eine umfangreiche Ausbildungs- und Weiterbildungskomponente, ein Glossar mit mehr als 900 Begriffen und eine umfangreiche Literatursammlung.

Von 1996 bis 1998 erstellte er eine Multimedia Show auf dem Gebiet der Historie der Computerentwicklung. Dieses Werk enthält u.a. ein 300-seitiges Hypertextsystem über die Computerentwicklung in Deutschland (Konrad Zuse), den USA (Aiken, Mauchly, Atanasoff, usw.) und England (u.a. Colossus, Baby, Mark I, usw.) und eine Simulation des ersten programmierbaren, auf dem binären Gleitkommasystems arbeitenden Rechners Z3 (Konrad Zuse).

Vom Februar 1998 bis Mai 1998 war er als Visiting Professor an der *University of Southwestern Louisiana* in Lafayette / USA tätig.

Am 9. Dezember 1998 hat der Fachbereichsrat Informatik der TU-Berlin ihm im Rahmen des Habilitationsverfahrens die Lehrbefähigung auf dem Gebiet der *Praktischen Informatik* zuerkannt.