



Title: Reverse Engineering Z4
Author(s): David Anders
Date: 15.03.10
Published by: Konrad Zuse Internet Archive
Source: Essay - ZIA ID: 0746

The Konrad Zuse Internet Archive preserves and offers free access to the digitized original documents of Konrad Zuse's private papers and to other related sources.

The Konrad Zuse Internet Archive is a nonprofit service that helps scholars, researchers, students and other interested parties discover, use and build upon a wide range of content in a digital archive. For more information about the Konrad Zuse Internet Archive, please contact zusearchive@zib.de.

Your use of the Konrad Zuse Internet Archive indicates your acceptance of the Terms & Conditions of Use (<http://zuse.zib.de/tou>) including the following license agreement. If you do not accept the Terms & Conditions of Use you are not permitted to use the material.

This work by Konrad Zuse Internet Archive is licensed under a
Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License
(<http://creativecommons.org/licenses/by-nc-sa/3.0/>).
Based on a work at <http://zuse.zib.de>



Attribution (BY) - You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work). Attribute with "Konrad Zuse Internet Archive (<http://zuse.zib.de>)".

Noncommercial (NC) - You may not use this work for commercial purposes.

Share Alike (SA) - If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

The usage of this document requires the consideration of possible third party copyrights, and might necessitate obtaining the consent of the copyright holder. The Konrad Zuse Internet Archive assumes no liability with respect to the rights of third parties. The Konrad Zuse Internet Archive is not responsible for the claims of any third party resulting from any infringement of copyright laws.



Arbeitsgruppe für künstliche Intelligenz

Diplomarbeit

im Studiengang Informatik

Reverse Engineering Z4

David Anders

Betreuer: Prof. Raúl Rojas

15. März 2010

Zusammenfassung

Die vorliegende Arbeit rekonstruiert und analysiert die Schaltungen und Funktionsweise der von Konrad Zuse entwickelten Rechenmaschine Z4. Die Schaltungen sind dem Dokument „Schaltzeichnungen für das Gerät Z4“, aus dem Jahr 1944, entnommen[1]. Eine große Hilfe bei der Analyse waren die „Notizen zu den Schaltungen für das Gerät Z4“, welche ebenfalls von Zuse verfasst wurden[2]. Die Z4 ist, wie die anderen Versuchsmodelle der Entwicklungsreihe Z1-Z5, ein Einzelstück und sollte der letzte Prototyp einer automatischen Rechenmaschine vor der Serienfertigung sein. Das Vorgängermodell Z3, wird als erster funktionierender Computer in der Geschichte gewertet[15]. Die Z4 beruht auf den bereits in den Vorgängermodellen eingesetzten Schaltungen und verbindet damit die Erfahrung mit neuen Weiterentwicklungen.

Es wird eine Übersicht über den Aufbau der verschiedenen Komponenten von der Ein- und Ausgabe, dem Speicher, dem Rechenwerk und Planwerk, sowie einiger weiterer Schaltungen gegeben. Eine detaillierte Analyse der Blockarchitektur, sowie der verwendeten Algorithmen geben ein vollständiges Bild der Z4. Schaltungen aus dem Original werden rekonstruiert und hinsichtlich der Funktionsweise analysiert und erläutert.

Inhaltsverzeichnis

1	Einleitung	5
2	Motivation von Zuse	7
3	Erste Rechenmaschinen	8
4	Z4 Übersicht	11
4.1	Struktureller Aufbau	13
4.2	Mechanischer Antrieb	16
4.3	Darstellung von Gleitkommazahlen	18
4.4	Befehlssatz	21
5	Blockarchitektur der Z4	24
5.1	Rechenwerk	24
5.2	Leitwerk	27
5.3	Planwerk	31
5.3.1	Tastatur - Planwerk	33
5.3.2	Pb - Locher	38
5.4	Eingabe von Dezimalzahlen	40
5.4.1	Eingabe von Zahlenwerten auf die Ta Relais	41
5.4.2	Anzeige der Eingabe auf dem Lampenfeld	46
5.5	Eingabe ins Dualsystem übersetzen	49
5.6	Lampenfeld	54
5.7	Architektur des Speichers	57
6	Arithmetische Algorithmen	64
6.1	Addition	64
6.2	Multiplikation	67
6.3	Division, Quadratwurzel, Multiplikation mit Pi	68
6.3.1	Division	68
6.3.2	Quadratwurzel	70
6.3.3	Multiplikation mit Pi	70
6.4	Hilfsoperation $\times 1/2$, $\times 2$, $\times 10$	71
6.5	x^2 und $1/x$	71

6.6	Sonderwerte	72
7	Fazit	74
	Literatur	76
A	Anhang	79
A.1	Lampenfeldanzeige	79
A.1.1	Lampenfeldanzeige durch Ta Relais	79
A.1.2	Weichenstraße Ta auf Tb Relais	80
A.1.3	Lampenfeldanzeige durch Tb Relais	81
A.2	Tastaturrelais	82
A.2.1	Ta Relais auf Rechenwerk / Tm Relais	82
A.2.2	Rechenwerk auf Tn / Ta Relais	82
A.3	Planwerkschaltungen	83
A.3.1	Belegung der Pb Relais durch Tastatureingabe	83
A.3.2	Befehlskreislauf Planwerk	84
B	Erklärung	85

1 Einleitung

Die vorliegende Arbeit rekonstruiert die Funktionsweise der von Konrad Zuse entwickelten Rechenmaschine Z4 und analysiert die verschiedenen Schaltungen. Die Z4 wurde in den Jahren 1941 bis 1945 entworfen und gebaut. Nach erster Fertigstellung im Jahr 1944, musste Zuse den Konstruktionsort Berlin verlassen und baute die Z4 im Jahr 1945 neu auf. Vor einem Verkauf an die ETH Zürich, im Jahr 1950, wurde die Z4 noch um die von der ETH geforderten Erweiterungen verändert. Der Einbau eines zweiten Abtasters, um Unterprogramme ablaufen zu lassen, war eine dieser Forderungen. Ihr Einsatz an der ETH Zürich dauerte bis 1954[5]. 1954 bis 1959 befand sich die Z4 im Institut Franco-Allemand des Recherches de St. Louis in Frankreich. Heute befindet sich die Z4 im Deutschen Museum in München.

Die Z4 ist ein Modell aus der Konstruktionsreihe Z1, Z2, Z3, Z4 und Z5 [6][14]. Technische Grundlagen einzelner Bauteile wurden zum Teil schon in den verschiedensten Schriften zu den Vorgängermodelle Z1, Z2 und Z3 aufgearbeitet[15]. Gemeinsamkeiten im logischen Aufbau sind somit beabsichtigt, da es sich bei der Z4 um eine Weiterentwicklung der bestehenden Prototypen handelt.

Die Grundlage für die Rekonstruktion der Arbeitsweise der Z4 wird durch die von Konrad Zuse hinterlassenen Dokumente, Patentanmeldungen und Quellen zu den bereits analysierten Bauteilen der Konstruktionsreihe und insbesondere der überlieferten Originalunterlagen zur Z4 gegeben. Hierbei bilden die „Schaltzeichnungen für das Gerät Z4“[2] und die „Notizen zu den Schaltungen für das Gerät Z4“[2] den Schwerpunkt. Weitere Dokumentationen aus der Zeit und bisher gewonnene Erkenntnisse der Vorgängermodelle komplettieren die Quellen.

Zu Beginn wird die Person Konrad Zuse vorgestellt. Ein Gefühl für die Umgebung und die damalige Situation von Konrad Zuse sollen einen Eindruck für die innovativen Entwicklungen geben. Vor diesem Hintergrund wird die Bedeutung, der aus heutiger Sicht trivialen Entwicklung, die notwendige Tragweite beigemessen. Insbesondere die Schwierigkeiten der Finanzierung und Beschaffung der Konstruktionsmittel während des 2. Weltkrieges, beweisen Zuse's ehrgeizigen Willen seine Vision Wirklichkeit werden zu lassen.

In Kapitel 3 findet sich eine kurze Übersicht der vorangegangenen Prototypen Z1 bis Z3, welche das Bild der Vorgänger komplettiert und die Entwicklung zwischen den Versuchgeräten deutlich macht. Die Beschreibung der Funktionsweise der Z4

beginnt mit einer allgemeinen Übersicht in Kapitel 4. Die einzelnen Komponenten des Versuchsgeräts werden kurz vorgestellt und die Funktionsweise beschrieben. Es folgt die Darstellung der Blockarchitektur in Kapitel 5, welche detailliert auf die einzelnen Komponenten und logischen Strukturen eingeht.

Auf die Erstellung von Rechenplänen und die Konstruktion des Planfertigungs- teils wird verzichtet, da diese bereits ausgiebig analysiert worden. Eine ausführliche Übersicht zur Erstellung der notwendigen Rechenpläne ist in „Die Architektur der Rechenmaschinen Z1 und Z3“[7] und „Rechenpläne für das Rechengerät V4 (Z4)“[13] zu finden.

Das Planfertigungsgerät ist für den Betrieb und die Analyse der verwendeten Schal- tung nicht erforderlich und stellt eine eigene Entwicklung dar. Eine Übersicht der Funktionsweise gibt das Dokument „Planfertigungsgeräte“[22][23].

Während der Konstruktionsphase hat Konrad Zuse nicht nur die Patentanmeldung, sondern auch weitere Schriften zur Z4 verfasst. Leider sind die durch das Patentamt überlieferten Unterlagen nicht vollständig. Seiten, welche den detaillierten Aufbau der beiden Additionswerke darstellen sind leider nicht mehr vorhanden. Die Addi- tionswerke sind in Ihrer Funktion denen der Z3 nachempfunden, weshalb bei der Betrachtung der Z4 von einer ähnlichen Arbeitsweise ausgegangen wird. Ein Teil der Schaltungen zur Sonderwertbehandlung ist überliefert, aber im Original nahezu unkenntlich, weshalb auch hier eine Analyse nur bedingt möglich ist.

Am Ende wird auf die verschiedenen Operationsbefehle und den damit verbundenen mathematischen und strukturellen Schaltungen eingegangen. Eine Analyse verschie- dene arithmetischer Algorithmen erfolgt in Kapitel 6. Ein Fazit in Kapitel 7 schließt die Arbeit.

Die Arbeit soll ein Verständnis für die Arbeitsweise der Z4 schaffen, ohne den Le- ser bemühen zu müssen, die Patentanmeldung oder weitere Schriften im Original zu studieren. Gleichzeitig soll das Einlesen in die originalen Unterlagen vereinfacht werden und die Grundlage geschaffen, das Schaltungen im Original schneller verstan- den und in den Gesamtzusammenhang gebracht werden können. Ein systematisches Verständnis der Vorgängermodelle Z1 und Z3 wird so nebenbei geschaffen, da der logische Aufbau der Z4 denen der Vorgängermodelle ähnlich ist. Eine Auswahl an Schaltungen wurde der Patentanmeldung entnommen und dieser Arbeit in digitali- sierter Form im Anhang und an zweckmäßiger Stelle beigelegt.

2 Motivation von Zuse

Konrad Zuse wurde am 22. Juni 1910 in Berlin geboren und verstarb am 18. Dezember 1995, in Hühnfeld, im Alter von 85 Jahren. Der Beginn seines Studiums gestaltete sich nicht geradlinig, da er als erstes den Studiengang Maschinenbau wählte, dann zur Architektur und schließlich zum Bauingenieurwesen wechselte, ohne in der Architektur oder dem Maschinenbau einen Abschluss zu erzielen. Seinen Abschluss als Bauingenieur erhielt er 1935[18]. Die Idee für eine automatische Rechenmaschine entwickelte Zuse schon während des Studiums. Immer wieder waren umfangreiche, sich im Ablauf wiederholende Rechnungen notwendig. Diese Rechnungen liefen immer nach einem ähnlichen Schema ab und erforderten einen hohen Zeitaufwand. Diese Rechnungen automatisch durchführen zu lassen und speziell den Bedürfnissen von Ingenieuren und Wissenschaftlern anzupassen, war Zuse's Vision[21]. Zuse erkannte, dass alle von ihm durchgeführten numerischen Rechnungen sich auf eine mathematische Grundform zurückführen lassen. Diese Grundform, von Zuse als "Rechenplan" bezeichnet, wird durch die Aneinanderreihung von einzelnen Basisoperationen zusammengesetzt [10][11]. Vergleichbar ist dieses mit der Darstellung von Nand und Nor Gattern, in der heutigen Aussagenlogik. Das von Zuse entwickelte Planfertigungsteil sollte es Wissenschaftlern ermöglichen, Rechenpläne für die Z4 zu erstellen, ohne sich über Gebühr mit der Programmierung der Z4 auseinander setzen zu müssen[13]. Das Planfertigungsteil sollte eine eingegebene mathematische Formel automatisch in einen Rechenplan umwandeln. Dieser Rechenplan konnte nun wiederum von der Z4 gelesen und mathematische Rechnungen effizient durchgeführt werden. Zuse wollte mit diesen Konstruktionen eine vollständige Automatisierung von mathematischen Rechnungen durchführen lassen.

3 Erste Rechenmaschinen

Zuse entwickelte während seines Lebens die verschiedensten Maschinen und meldete eine unglaubliche Anzahl an Patenten an[4]. Die Rechenmaschinen mit dem Namen Z1-Z4 erhielten diesen erst später. Während der Entwicklung wurden diese von Zuse treffend als V1-V4 benannt. Das V steht stellvertretend für Versuchsmodell. Die mit den Versuchsmodellen gemachten Erfahrungen und Erkenntnisse sollten die Grundlagen für eine spätere Serienfertigung bilden[4].

Die Z4 sollte der letzte Prototyp vor der Serienfertigung sein. Die Unterschiede zwischen Z3 und Z4 sind kleiner als zwischen den Vorgängermodellen geworden. Die logische Struktur und der Aufbau der Komponenten wurde weitestgehend beibehalten. Im ersten Moment erscheint der Schritt, einen verbesserten Speicher auf Grundlage der mechanischen Schaltgliedtechnik zu verwenden[16][17], wie dies bei der Z1 der Fall war, als nicht schlüssig. Es wird gezeigt, dass Zuse den mechanischen Speicher aus Kosten- und Konstruktionsgründen wählte. Grundlegende Änderungen zwischen Z4 und Serienfertigung waren nicht geplant. Auf Grund der schwierigen Umstände des Krieges und der sehr hohen Konstruktionskosten, ist es nie zur erträumten Serienfertigung gekommen. Dieses ist allerdings nicht zu letzt der Besetzung Deutschlands durch die Alliierten und daraus folgenden politischen und wirtschaftlichen Entscheidungen geschuldet.

Die ersten Entwicklungen von Zuse erfolgten auf rein mechanischer Basis, allerdings hier schon mit einer internen Darstellung der Zahlen im Dualsystem. Die gesammelten Erkenntnisse wurden in der mechanischen Schaltgliedtechnik verwirklicht. Ein mechanisches Schaltglied simuliert die Funktion eines Relais und kann die Werte Ja oder Nein annehmen[17]. Das Grundprinzip besteht darin, das mechanische Bleche in zueinander rechtwinkligen Positionen aufgebaut sind. Durch die explizite Stellung des Bleches, können dieses den Wert Ja oder Nein (0 oder 1) annehmen. Ein Steuerblech ermöglicht oder verhindert hierbei die Übertragung auf ein folgendes Blech. Ein Blech kann damit steuernde und nicht steuernde Wirkung haben. Aus solchen einfachen Schaltgliedern lassen sich unter Verwendung der Schaltalgebra komplexe Rechenwerke aufbauen, wie sie in den von Zuse entwickelten Rechengeräten zum Einsatz kamen.

Die Z1 ist das erste Modell der Versuchsreihe und wurde in den Jahren 1936-38

entwickelt und ist aus rein mechanischen Bauteilen konstruiert. Der Speicher war ebenfalls nur aus den soeben beschriebenen mechanischen Schaltungen aufgebaut. Eine detaillierte Beschreibung der Z1 findet sich in weiteren Dokumenten[9].

Das Nachfolgemodell der Z1 ist die Z2. Die Z2 wurde im Jahr 1940 konstruiert und besteht aus ca. 200 Relaisschaltungen. Für Rechnungen wurde in diesem Modell eine feste Kommastelle implementiert. Als Speicher dient der schon für die Z1 entwickelte mechanische Speicher mit 64 Speicherzellen. In der Z2 wird erstmalig ein Lochstreifen verwendet. Dieser besteht aus einem Filmstreifen, der durch entsprechende Lochungen einen automatisch ablaufenden Rechenplan ermöglichen sollte. Die Z2 ermöglichte es einfache Formeln durchzurechnen und das Prinzip einer programmierbaren Rechenmaschine darzustellen.

Die Konstruktionsprinzipien der Z2 sahen für Zuse nicht sehr vielversprechend aus, weshalb das nächste Versuchsmodell, die Z3, auf den logischen Strukturen und Schaltungen der Z1 aufbaut. Die Z3 wurde in den Jahren 1938 bis 1941 entwickelt und im Jahr 1941 beim Patentamt unter der Bezeichnung Z391 eingereicht[20]. Die wesentliche Veränderung zwischen Z3 und den Vorgängermodellen ist in den verwendeten Bauteilen zu finden. Die Z3 verwendet für die einzelnen Komponenten keine mechanischen Schaltglieder mehr, sondern hat diese durch Relais aus der Fernsprechtechnik ersetzt. Relais können ebenfalls den Wert 0 oder 1, angezogen oder nicht, annehmen. Komplexe Schaltungen lassen sich durch Relais einfacher realisieren, als es mit den mechanischen Bauteilen möglich gewesen wäre. Für die Z3 sind ca. 1400 Relais für den Speicher und 600 für den Prozessor verbaut worden. Das oft diskutierte Fehlen von bedingten Sprüngen wurde bereits in „Die Architektur der Rechenmaschinen Z1 und Z3“ [7] ausgiebig diskutiert. Ein universeller Einsatz der Z3 war grundsätzlich möglich[19].

Die Z3 arbeitete als Prototyp zuverlässig und sollte somit dem Nachfolger, der Z4 den Weg für die Serienfertigung ebnen. Die Entwicklung der Z4 erfolgte in den Jahren 1941 bis 1945. Hierbei wurde die logische Struktur der Z3 weiter verwendet. Auf die Gemeinsamkeiten und Unterschiede wird im weiteren Verlauf eingegangen. Hier ist allerdings schon zu erwähnen, dass die Z4 keinen Speicher aus Relaischaltern,

sondern einen mechanischen, wie er bereits bei der Z1 zum Einsatz kam, verwendete. Wäre der Speicher mit Relais realisiert worden, wären mehr als 32000 Stück davon notwendig gewesen, da die Z4 einen Speicher mit 1024 Speicherzellen erhalten sollte[24]. Die Verwendung von Relais hätte den Speicher zu groß, zu schwer und zu teuer werden lassen. Der eigene von Zuse entwickelte mechanische Speicher, wie er schon bei der Z1 im Einsatz war, war kostengünstiger und einfacher zu realisieren. Der in der Z4 verwendete größere Adress- und Speicherraum und der größere Befehlssatz sind nur einige Neuerungen, auf welche in den folgenden Kapiteln näher eingegangen wird. Viele der in der Z4 verwendeten Operationen waren bereits in der Z3 vorhanden, dem Programmierer aber verborgen. In der Z4 wurden diese Befehle dem Programmierer zugänglich gemacht, dass diese direkt angesprochen werden können. Dies erhöhte den universellen Einsatz der Z4, ohne das grundlegende Änderungen am Design vorgenommen werden mussten. Zusätzlich erhöhte dieses die Vermarktungschancen erheblich.

1945 wurde die Z4 fertig gestellt, auf Grund der Materialknappheit auf die Grundfunktionen beschränkt und nur mit einem 64 Zellen umfassenden Speicher ausgestattet. Nach Ende des Krieges wurde die Z4 an Prof. Stiefel an die Eidgenössische Technische Hochschule Zürich verkauft und war dort knapp fünf Jahre im Einsatz. Voraussetzung hierfür war die Erweiterung der Z4 um einige weitere Funktionen, welche in der allgemeinen Beschreibung der Z4 erstmal keine Beachtung finden werden. Das eine Erweiterung des Speichers und der Funktionen nach Fertigstellung möglich waren, lässt hier schon deutlich werden, wie universell und vorausschauend Zuse die Konstruktion erdacht hat.

Die Gemeinsamkeiten der Entwicklungsreihe Z1-Z4 sind beachtlich. Ursache ist im Speziellen, der ständige Weiterentwicklungsdrang von Konrad Zuse. Während der Entwicklung und Fertigstellung eines Versuchsgerätes wurden von ihm bereits neue Ideen und Verbesserungsmöglichkeiten kreiert und die ersten Entwürfe für das folgende Versuchsgerät erstellt. Als durchgängige Gemeinsamkeit lässt sich die Eingabe im Dualsystem nennen. Die Eingabe erfolgte im Dezimalsystem, wobei die Umformung in das Dualsystem automatisch erfolgte. Als besonders wegweisend erschloss sich die Verwendung der halblogarithmischen Darstellung der Zahlen, auf welche noch näher eingegangen wird.

4 Z4 Übersicht

In diesem Kapitel wird ein erster Überblick über die Funktionsweise und den strukturellen Aufbau der Rechenmaschine Z4, im Folgenden nur Z4 genannt, gegeben. Es wird mit einer allgemeinen Beschreibung der Z4 begonnen, welche den strukturellen Aufbau erklären und einen ersten Eindruck für die verschiedenen Komponenten geben soll. Eine detaillierte Darstellung der Blockarchitektur erfolgt im nächsten Kapitel.

Die Z4 ist das direkte Nachfolgemodell der Z3 aus dem Jahr 1938[7]. Die Entwicklung der Z4, welche im Wesentlichen eine verbesserte Z3 ist, erfolgte schon während der Konstruktion der Z3. Während Zuse weitere Erfahrungen mit der Konstruktion der Z3 sammelte, wurden die ersten Pläne für die Z4 entworfen. Viele Komponenten der Z4 lassen sich in der Z3 wieder finden. Letzte Verbesserungen und kleine Änderungen sollten die Z4 zum letzten Prototypen vor der geplanten Serienfertigung werden lassen.

Der Einsatz der Z4 in Serie war für kleine bis große Unternehmen und Forschungseinrichtungen geplant. Die Z4 sollte arbeitsintensive, monotone, fast gleich ablaufende Rechnungen automatisch durchführen. Ein Vertrag mit der Deutschen Versuchsanstalt für Luftfahrt (DVL) ermöglichte Zuse die Finanzierung. Als Auftraggeber der Z4 hatte die DVL natürlich eigene Anforderungen und Schwerpunkte[3]. Zuse, der nicht nur Erfinder, sondern auch kaufmännisches Geschick besaß, wusste um die wichtige Bedeutung einer funktionierenden Rechenmaschine für seinen Auftraggeber. Hierfür wurden aus Kosten- und Zeitgründen nicht alle der für die Z4 geplanten Funktionen implementiert. Deutlich wird dieses beim Befehlssatz, welcher in der ersten Ausbaustufe nicht alle Befehle zur Verfügung hatte. Dieses musste Zuse in Kauf nehmen, da die Entwicklung und Konstruktion sehr kostenintensiv war. Die Maschine wurde auf die Bedürfnisse der DVL zugeschnitten. Als Beispiel lässt sich hierfür die Implementierung der bedingten Sprünge erwähnen. Diese waren Zuse durchaus bekannt, für eine Abnahme der Maschine durch die DVL aber nicht zwingend erforderlich und wurden daher auch nicht implementiert. Anders sieht es nach dem Krieg aus, als die Z4 an die ETH in Zürich verkauft werden soll. Bedingte Sprünge sind hier eine Voraussetzung gewesen, weshalb diese relativ einfach von Zuse mit dem Einbau eines zweiten Abtasters realisiert wurden.

Die größten Unterschiede zwischen Z3 und Z4 sollen kurz angesprochen werden. Der wesentlich größer geplante Speicher mit 1024 Speicherplätzen und einer Bitlänge von 32 Bit war dem Speicher der Z3 deutlich überlegen. Die Z3 hatte nur 64 Speicherzellen mit jeweils 24 Bit zur Verfügung. Auch wenn in der ersten Ausbaustufe der Z4 nur 64 Speicherzellen realisiert wurden, war eine Erweiterung aber ohne Probleme möglich. Der Speicher ist modular aufgebaut und Zuse hat alle technischen Voraussetzungen in der Z4 implementiert, um diesen mit 1024 Speicherzellen zu betreiben. Die Erweiterung von 24 auf 32 Bit ermöglicht im Wesentlichen eine höhere Genauigkeit der gespeicherten Zahlenwerte, da die zusätzlichen Bit für mehr Stellen der Mantisse verwendet werden.

Die Art des Speichers unterscheidet sich auch zur Z3. Der Speicher der Z3 wurde mit Relais, wie für das Rechenwerk realisiert. Bei 1024 Speicherzellen mit jeweils 32 Bit, wären damit mehr als 32.000 Relais benötigt worden. Diese hätten nicht nur eine Menge physikalischen Platz benötigt, sondern hätten auch extrem hohe Kosten verursacht. Eine Beschaffung von 32.000 Relais war während der Kriegsjahre auch alles andere als einfach. Daher erinnerte sich Zuse an die Speicherkonstruktion der Z1. Diese verwendete einen von Zuse selbst entwickelten mechanischen Speicher, auf Grundlage der mechanischen Schaltgliedtechnik. Ein geringeres Volumen und kostengünstigere Produktion waren hier die Hauptgründe. Die Funktionsweise des mechanischen Speichers wurde bereits mehrfach analysiert[8][9], weshalb in dieser Arbeit vornehmlich auf das Ansprechen des Speichers durch das Plan- und Rechenwerk eingegangen wird.

Die Ein- und Ausgabe der Z4 ist wie bei der Z3 über die angeschlossene Tastatur und das Lampenfeld zur Anzeige der Rechenergebnisse realisiert. Über einen Abtaster, welcher einen auf einem Filmstreifen hinterlegten Rechenplan abarbeitet, kann die Eingabe ebenfalls erfolgen. Die Rechenplananfertigung kann über das Planfertigungsteil erfolgen. Das Planfertigungsteil ist als eigenständige Entwicklung zu sehen und für den Betrieb der Z4 nicht notwendig. Absicht von Zuse war es, mit Hilfe des Planfertigungsteils einfachen Wissenschaftler, welche mit der Programmierung der Z4 nicht vertraut sind, eigene Rechenpläne entwickeln zu lassen. Eine kurze Einführung von einem halben Tag, soll die Bedienung des Planfertigungsteils ermöglicht haben[5]. Um den Betrieb der Z4 effizienter zu gestalten, ist das Planfer-

tigungsteil in die Konstruktion der Z4 angegliedert.

Der für den Anwender nutzbare Befehlssatz ist in der Z4 deutlich größer, als er noch in der Z3 war. Im Wesentlichen sind viele der bereits in der Z3 implementierten Befehle dem Bediener zugänglich gemacht. Als Beispiel dient die Rechenoperation „Multiplikation mit 10“. Diese wird bereits in der Z3 verwendet, um die Übersetzung vom Dezimal- ins Dualsystem durchzuführen. Ein gezieltes Ansprechen dieses Befehls ist vom Bediener allerdings erst in der Z4 möglich.

4.1 Struktureller Aufbau

Die Rechenmaschine Z4 hat, wie ihre Vorgängermodelle, die Entwicklung moderner Rechengeräte weit voran gebracht. Im 20. Jahrhundert wurden die ersten Grundsteine für die heutige PC Architektur gelegt. Die Z4 besteht aus Rechenwerk, Speicherwerk, Planwerk, Kommando- und Ablesevorrichtung (die Ein- und Ausgabe). Damit findet sich eine Realisierung der von-Neumann-Architektur in leicht abgewandelter Form, in der Z3 und Z4.

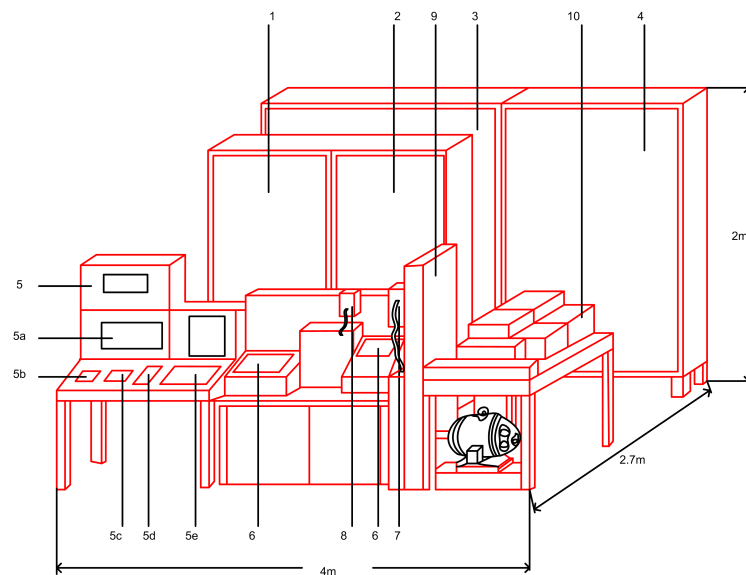


Abbildung 1: Zeigt den strukturellen Aufbau der Z4.

In der Abbildung 1 lässt sich die bereits angesprochene Trennung der einzelnen Komponenten deutlich erkennen. 1-4 sind Relaisschränke, welche die mehr als 2000 Relais aufnehmen. In diesen befinden sich die Schaltungen der Additionswerke, dem

Leitwerk, dem Planwerk und dem Sonderwerk. Unter Punkt fünf ist das gesamte Schaltpult zusammengefasst. Dieses ist wiederum in die Resultatanzeige (5a), das Protokoll (5b), die Tastatur für die Zahlen (5c), für die Operationen (5d) und für die Speicherzellenanwahl(5e) aufgeteilt ist. Die Rechenplankontrolle (6) findet sich ebenso, wie der Locher (7) und Abtaster (8). Das Getriebe (9), welches über den Motor die einzelnen Schaltungen antreibt und synchron hält, sowie das Speicherwerk (10) in der ersten Ausbaustufe, sind ersichtlich.

Die Ein- und Ausgabe ist jeweils als eine Komponente zu sehen, welche wiederum in Ein- und Ausgabe getrennt wird. Die Eingabe erfolgt über die angeschlossene Tastatur und ermöglicht die Eingabe von Zahlenwerten, Speicheradressen und Operationsbefehlen. Die Ausgabe erfolgt über das Lampenfeld zur Ausgabe. Der Prozessor welcher in Form eines Rechenwerks realisiert ist, wird durch das Leitwerk gesteuert. Der mechanische Speicher ist, mit bis zu 1024 Zellen, über Datenleitungen mit dem Rechenwerk verbunden. Die Steuerung und das Zusammenspiel der verschiedenen Komponenten werden durch das Planwerk gesteuert. Die Befehle werden im Planwerk weiterverarbeitet und auf die einzelnen Komponenten übertragen.

Die Z4 besteht aus ca. 2200 Relais. 21 Schrittschalter sprechen die verschiedenen Steuerrelais an[12]. Jeder Schrittschalter führt jeweils ein Mikroprogramm durch. Durch die Abfolge eines Schrittschalters, führt dieser in jedem neuen Takt einen weiteren Schritt aus und aktiviert so nacheinander festgelegte Relais. Komplizierte Operationen, wie die Division, werden so automatisch gesteuert.

Die Ein- und Ausgabe erfolgt über die Kommando- und Ablesevorrichtung, sowie dem Lampenfeld zur Ausgabe. Die Eingabe kann über die Tastatur oder den angeschlossenen Abtaster zur automatischen Steuerung, durch einen Rechenplan, erfolgen. Die eingegebenen Befehle werden auf das Planwerk übertragen, welches die weiteren Schritte steuert.

Erfolgt die Eingabe über die Tastatur, müssen Zahlenwerte im Dezimalsystem und die passenden Rechenoperationen kommandiert werden. Hierbei ist zu beachten, dass zuerst Operand 1, Operand 2 und dann die Rechenoperation eingetastet werden muss. Während der Eingabe wird die eingegebene Zahl auf dem Lampenfeld angezeigt und ermöglicht dem Bediener eine Korrektur bei falscher Eingabe. Eine

Korrektur während der Eingabe ist nur möglich, solange die Zahl noch nicht auf das Rechenwerk übertragen wurde. Die Übertragung von den Tastaturrelais auf das Rechenwerk wird durch die Fertigtaste kommandiert.

Erfolgt die Eingabe über den Abtaster, erfolgt die Abfolge der Befehle automatisch. Der Benutzer muss bei entsprechender Programmierung nur die jeweiligen Variablen eintasten und so die Operanden für die Rechnungen eingeben. Die Abfolge des Rechenplans führt komplizierte mathematische Rechnungen durch, welche aus den Basisoperationen zusammengesetzt sind. Der Abtaster ließt einen 36mm breiten Filmstreifen ein, welcher bis zu vier Löcher pro Zeile aufweisen darf. Ein Befehl ist immer durch zwei Zeilen, also bis zu acht Lochungen codiert. Durch die Lochung sind bis zu 2^8 mögliche Codierungen möglich. Wird ein Speicher mit mehr als 64 Zellen verwendet, muss auf zwölfstellige Befehle umgeschaltet werden. Die zusätzlichen vier Lochungen befinden sich in der nächsten Zeile des Lochstreifens. Pro Sekunde können bis zu fünf Operationen vom Abtaster eingelesen werden.

Der erste Rechenplan auf dem Filmstreifen muss über die Tastatur auf den Locher gegeben werden, welcher den Filmstreifen entsprechend locht. Ein Rechenplan ist eine Aneinanderreihung von einzelnen Befehlen und Speicheradressen. Ist dieser Rechenplan einmal fertig gestellt, kann dieser beliebig oft als Grundlage für weitere Rechnungen genutzt werden. Die Eingabe kann von der Tastatur direkt auf den Lochstreifen gegeben werden oder erfolgt durch den Einsatz des Planfertigungsgeräts. Die Befehlseingabe in die Z4 kann natürlich auch von der Tastatur erfolgen, ohne einen Lochstreifen zu produzieren. Die Anzeige von Zwischen- und Endergebnissen erfolgt über verschiedene Lampen- und Anzeigevorrichtungen, auf welche noch im Einzelnen eingegangen wird.

Das Rechenwerk beinhaltet mehrere Register zur Aufnahme der Operanden und Zwischenergebnisse und führt die Rechenoperationen Addition und Subtraktion durch. Alle anderen Rechenoperationen werden durch eine geschickte Abfolge von Additions- und Subtraktionsbefehlen realisiert. Hierfür sind zwei Additionswerke A und B, für die im Dualsystem dargestellte Zahl vorgesehen. Die Exponenten werden im Additionswerk A und die Mantissen im Additionswerk B verarbeitet.

Die Grundrechenoperationen können durchgeführt werden, sobald der oder die Operanden in den Registern vorliegen. Eine Eingabekontrolle lässt die Rechnungen nur

starten, wenn für Operationen mit einem Operand dieser bereits eingegeben wurde und für Rechnungen mit zwei Operanden bereits beide Operanden auf den Registern im Rechenwerk liegen. Dem Rechenwerk gehören noch mehrere Weichenstraßen und Steuerelemente an, welche im Kapitel der „Blockarchitektur der Z4“ detailliert analysiert werden.

Das Speicherwerk besteht im Gegensatz zur Z3 wieder aus rein mechanischen Bauteilen, wie es schon bei der Z1 der Fall war. Es nimmt auf Befehl die Ausgangs-, Zwischen- und Resultatwerte einer durchgeführten Rechnung auf. Die Z4 ist für ein Speicherwerk mit bis zu 1024 Stellen ausgelegt. Zum Ende des zweiten Weltkrieges besitzt die Z4 allerdings nur 64 Speicherzellen. Dieses ist für die reine Funktionsweise ausreichend und der Speicher kann jederzeit auf bis zu 1024 Speicherzellen erweitert werden. Pro Speicherzelle kann ein Wert mit einer Länge von 32 Bit aufgenommen werden. Jede Zelle des Speicherwerks kann direkt und einzeln vom Rechenwerk und vom Planwerk angesprochen werden. Das Speicherwerk ist so mit dem Rechenwerk verbunden, dass das Rechenwerk einen Wert auf eine explizite Speicherzelle legen kann und auch den Wert jeder Speicherzelle auslesen kann. Die Übertragung von Werten erfolgt in beide Richtungen. Eine detaillierte Beschreibung des Speichers erfolgt in Kapitel 5.7.

Die Koordinierung der Befehle und Aktivierung der verschiedenen Steuerrelais erfolgt durch das Planwerk. In diesem werden die Befehle und Zahlenwerte der Eingabe verarbeitet und auf die richtigen Komponenten verteilt. Das Planwerk steuert das Zusammenspiel und die Übertragung der Eingaben und Operationsbefehlen zwischen den verschiedenen Registern. Die Eingabe von Zahlenwerten kann auf den Speicher oder das Rechenwerk geleitet werden. Operanden im Planwerk lösen die Abfolge der Rechenbefehle und Aktivierungen im Leitwerk aus, welches damit das Rechenwerk steuert.

Die Steuerung von Locher und Abtaster erfolgt ebenfalls durch das Planwerk, wobei die Eingabe über den Abtaster zu Kontrollzwecken manuell überbrückt werden kann.

4.2 Mechanischer Antrieb

Die Z4 wird durch einen Elektromotor mit einer Leistung von mehr als 1400 Umdrehungen pro Minute angetrieben. Dieser treibt den elektrischen Impulsgeber, den

mechanische Impulsgeber des Speicherwerks und die Antriebswelle für Abtaster und Locher an. Für einen zuverlässigen Betrieb sind allerdings nur 300 Umdrehungen pro Minute notwendig, weshalb der Elektromotor untersetzt wird. Durch die Maßeinheit von fünf Umdrehungen und damit gleichzeitig Takten pro Sekunde, kann der Z4 eine Rechengeschwindigkeit von 5Hz attestiert werden. Dieses entspricht in etwa dem Vorgängermodell Z3.

Der elektrtrische Impulsgeber besteht aus einer Schaltwalze, welche in jeder Umdrehung fünf Impulse an die Z4 gibt. Jede Umdrehung wird von Zuse als ein Spiel bezeichnet, welches wiederum in fünf Impulse aufgeteilt ist. Dafür sind auf der Walze fünf leitende Bereiche mit jeweils 72° eingerichtet. Zwischen jedem leitenden Bereich befindet sich ein isolierter Bereich, welcher den Impuls unterbricht.

Die Unterteilung in Ansprech- und Halteimpulse hat einen leitenden Bereich von etwas weniger als 72° bei den Ansprechimpulse und etwas mehr als 72° bei den Halteimpulsen zur Folge. Es kann somit immer nur ein Ansprechimpuls gleichzeitig aktiviert sein und zwischen den Halteimpulsen entsteht keine Unterbrechung. Die Relais können damit nicht abfallen.

Durch die Aktivierung von maximal 25 Impulsen pro Sekunde, darf ein Impuls maximal 40ms andauern. Die verwendeten Relais müssen daher in der Zeit ansprechen und auch schnell genug wieder abschalten. Eine höhere Rechengeschwindigkeit würde schneller schaltende Relais benötigen.

In der Regel benötigen die von Zuse verwendeten Relais 30ms zum schalten und 60ms zum abschalten. Daher wurden Impulse mit vier Schritten vermieden. Ansprechimpulse Schalten das Relais. Halteimpulse sollen das Abfallen eines Relais verhindern und diesem die Möglichkeit geben im nächsten Schritt steuernd zu wirken.

Wenn für die Durchführung einer Operation keine Relaiskette, sondern kompliziertere Schaltungen notwendig sind, werden Schrittschalter verwendet. Diese schalten in jedem Takt einen Schritt weiter und liegen permanent an Strom, sobald dieser von der Grundstellung weiter geschaltet ist. Um ein Durchbrennen des Schrittschalters zu verhindern, läuft dieser in jedem Spiel selbstständig weiter, bis er wieder die Grundstellung erreicht. In jedem Schritt kann dann eine Schaltung von verschiedenen steuernden Relais realisiert werden. Die Schrittschalter entsprechen damit mehr oder weniger einer verdrahteten Mikroprogrammierung, zur Abfolge von Mikroprogrammen wie der Multiplikation.

4.3 Darstellung von Gleitkommazahlen

Konrad Zuse verwendet als einer der Ersten die halblogarithmische Zahlendarstellung im Dualsystem. Der größte Vorteil der Verwendung des Binärsystems besteht in der Verwendung der benötigten Bauteile. Für die Darstellung des Dezimalsystems könnte jede Ziffer den Wert 0-9 einnehmen, im Dualsystem nur den Wert 0 oder 1. Bauteile, welche nur den Wert 0 oder 1, wahr oder falsch anzeigen, sind viel einfacher zu beschaffen, als Bauteile, die den Wert 0-9 annehmen müssen. Die Konstruktion und Verwendung von Bauteilen, welche nur zwei Werte einnehmen können, reduziert die Fehleranfälligkeit in der Verdrahtung und Konstruktion der logischen Schaltungen. Jede mathematische Formel lässt sich mit der „Aussagenlogik“ durch drei Basisoperationen darstellen. Mit den Basisoperationen: Und, Oder und Negation lassen sich alle mathematischen Formeln darstellen[26]. ein Bauteil, welches diesen Anforderungen gerecht wird, war relativ einfach zu finden. Zuse verwendet einfache Relais wie sie im Fernsprechbau verwendet wurden. Diese konnten angezogen oder nicht angezogen sein und damit den Wert 0 oder 1 repräsentieren. Eine interne Darstellung im Dezimalsystem, die vorhandenen Bauteile vorausgesetzt, hätte zudem deutlich kompliziertere Schaltungen erfordert.

Bei Rechenmaschinen mit feststehendem Komma, muss die Größenordnung des Ergebnisses vorher bekannt sein. Bei den in dieser Zeit weit verbreiteten Rechenmaschinen auf Lochkartenbasis, war das Komma fest gesetzt. Sollten die Operanden sehr groß und sehr klein sein kommt es schnell zu einem Über- oder Unterlauf und das Ergebnis der Rechnung ist nicht verwertbar. Durch die Verwendung der halblogarithmischen Darstellung und der Möglichkeit ein gleitendes Komma zu verwenden, können Operationen mit sehr großen und sehr kleinen Zahlen in einer Rechenoperation Anwendung finden.

Im Dezimalsystem wird bei der halblogarithmischen Darstellung die Zahl 19 in der Form $19 = 10^1 * 1,9$ repräsentiert. Im Dualsystem mit $19 = 2^4 * 1,1875$. Der Exponent besitzt den Wert vier und die Mantisse den Wert 1,1875. Es wird sofort deutlich, dass für die Darstellung im Dualsystem deutlich mehr Stellen für dieselbe Zahl benötigt werden. In der Regel sind 3-4-mal mehr Stellen notwendig, um dieselbe Zahl zu repräsentieren. Die Zahl vor dem Komma der Mantisse ist immer eine 1

und muss daher nicht explizit gespeichert werden. Damit werden nur die Stellen der Mantisse hinter dem Komma im Speicher der Z4 abgelegt[25].

Die Anzahl der Stellen der Mantisse, gibt deren Genauigkeit an. Je mehr Stellen vorhanden sind, desto genauer können die Zahlen dargestellt werden bzw. desto kleiner werden die Zwischenräume zwischen zwei aufeinanderfolgende Zahlen im Zahlenraum.

Bei einer Mantisse mit vier Stellen ist die nächst größere darstellbare Zahl 19,0016. $2^4 * 1,1875 = 19$ und $2^4 * 1,1876 = 19,0016$. Zahlen zwischen 19,0000 und 19,0016 lassen sich bei einer Mantissenlänge von vier, nicht darstellen und es kann zu Rundungsfehlern kommen. Bei einer Mantisse mit fünf Stellen $2^4 * 1,18750 = 19$ und $2^4 * 1,18751 = 19,00016$ ist die Genauigkeit entsprechend größer.

Fehler während der Rechenoperationen können insbesondere dann auftreten, wenn zwei fast gleich große Zahlen voneinander abgezogen werden und so der Wert Null herauskommt. Wäre die Genauigkeit größer könnte das Ergebnis eine sehr kleine Zahl werden, ohne den Wert Null anzunehmen. Gerade bei Zwischenergebnissen kann so die gesamte Rechnung ungültig werden. Bei der Subtraktion oder Division einer besonders kleinen Zahl mit einer großen würde die besonders große Zahl nicht verändert werden.

Die Eingabe von Zahlen, als Operanden für die Rechnungen, erfolgt im Dezimalsystem. Hierfür werden die einzelnen Zahlen über die Tastatur eingegeben. Die Eingabe wird in der halblogarithmischen Form mit Alpha für den Exponenten und Beta für die Mantisse: $y = 10^a * b$ gehalten. Die Umformung in das Dualsystem, bei Übergabe der eingetasteten Zahl auf das Rechenwerk, erfolgt automatisch durch einen fest verdrahteten Ablauf verschiedener Befehle. Die Darstellung im Dualsystem folgt der Form $y = 2^a * b$.

Im Speicher der Z4 sind 32 Stellen für die Speicherung der Zahlenwerte vorgesehen. Die Aufteilung der 32 zur Verfügung stehenden Bit erfolgt in Exponent, Mantisse und Sonderwerte. Alpha beeinflusst die Größe des Zahlenbereichs und die Genauigkeit der zu verarbeitenden Zahlen wird durch die Mantisse festgelegt. Je mehr Stellen diese besitzt, desto genauer kann eine Zahl dargestellt werden. Je größer Alpha ist, desto größer können die verwendeten Zahlen sein.

Der Speicher besitzt 32 Speicherstellen. Die 32 Bit sind aufgeteilt in sieben Bit für

den Exponenten, 22 Bit für die Mantisse und jeweils ein Bit für das Vorzeichen, das Anzeigen von imaginären Werten und Sonderwerte. Der Exponent Alpha ist ganzzahlig und die Mantisse Beta erfüllt die Bedingung $1 \leq b < 2$. Die Mantisse wird als normierte Mantisse bezeichnet, weil sie die Bedingung $1 \leq b$ erfüllt und damit nie den Wert Null annehmen kann. Der Exponent wird im Zweierkomplement dargestellt. Negative Alpha können durch eine 1 auf der ersten Stelle ausgemacht werden. Ist die erste Stelle eine 0, ist Alpha positiv oder Null. Bei einem Exponenten mit sieben Bit können Alpha von -64 bis +63 dargestellt werden. Die normierte Mantisse besitzt 22 Stellen hinter dem Komma. Vor dem Komma befindet sich immer eine 1, welche nicht extra abgespeichert werden muss. Problematisch ist in dieser Darstellung, dass die Mantisse den Wert Null nicht annehmen kann. Die Darstellung der Zahl Null muss durch eine Ausnahmeregel erfolgen. Ein Exponent von -64 wird im vorliegenden Fall als Null interpretiert. Besonders große und besonders kleine Zahlen müssen ebenfalls durch eine Ausnahmeregel angezeigt werden. Unendlich wird daher durch einen Exponenten von +63 symbolisiert.

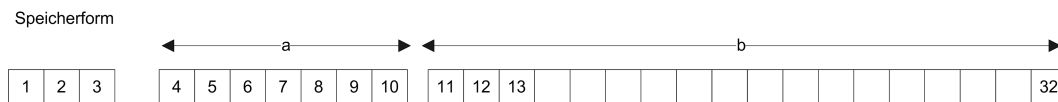


Abbildung 2: Zahlendarstellung im Speicher.

Auf der Stelle eins wird angezeigt, ob es sich bei der Zahl um eine Imaginärzahl handelt. Stelle zwei zeigt das Vorzeichen der Zahl an und Stelle drei, ob es sich ggf. um Sonderwerte der Form „+Unendlich“ oder „-unendlich“, handelt.

Die kleinste positive speicherbare Zahl ist somit $2^{-63} * 1,00000 \approx 1,0845 * 10^{-19}$ und die größte $2^{62} * 1,99999 \approx 9,22 * 10^{18}$. 2^{63} ist unendlich und 2^{-64} Null. Dieses entspricht dem Zahlenbereich der Z3. Der Exponent ist dort ebenfalls durch sieben Bit angegeben. Die zusätzlichen Stellen in der Z4, finden Ihre Verwendung in einer Mantisse mit 22 Stellen. Die Z3 hat nur 14 Stellen der Mantisse gespeichert. Durch die weiteren Stellen der Mantisse sind die Zahlenzwischenräume verringert und die Genauigkeit der darstellbaren Zahlen verbessert. Der Exponent in der dezimalen Darstellung wird durch das Drücken der Taste -6 oder +6 verändert. Jede der beiden Tasten kann bis zu dreimal gedrückt werden. Eingaben für den Exponenten sind somit von -18 bis +18 möglich. Der Bediener muss darauf achten, dass keine

Zahl größer $9,22 * 10^{18}$ eingetastet wird, da diese nicht mehr darstellbar ist. sollte es doch zu einer Eingabe außerhalb des Zahlenbereichs kommen, wird dieses durch eine Warnlampe angezeigt. Das Komma ist gleitend und kann sich bei der Eingabe an jeder Stelle befinden.

Die Register des Rechenwerks sind noch um einige Stellen erweitert. Die Register im Additionswerk A, für den Exponenten, besitzen acht Bit. Gespeichert werden allerdings nur sieben Bit. Das Additionswerk B greift auf Register mit bis zu 29 Stellen zu. Hierbei liegen bis zu drei Bit vor und 26 Stellen hinter dem Komma. Die zusätzlichen Stellen dienen auch hier für eine erhöhte Genauigkeit der Rechnungen und um die Gefahr von Überläufen zu reduzieren.

4.4 Befehlssatz

Mit einer Codierung von acht Bit lassen sich bis zu 256 mögliche Befehle codieren. Die Z4 kann allerdings nur knapp 50 verschiedene Befehle verarbeiten, was im Vergleich zum Vorgängermodell der Z3[10], mit neun Befehlen eine deutliche Verbesserung ist. Die Tastung der Befehle erfolgt in der Regel über die Operationstasten der Tastatur oder durch das Einlesen eines codierten Lochstreifens. Die Eingabe von Hand über die Tastatur dauert deutlich länger, als das Einlesen des Lochstreifens mit bis zu fünf Operationen pro Sekunde. Daher wird die Eingabe über Tastatur üblicherweise nur gewählt, wenn die Rechnung so trivial ist, dass die Erstellung eines Rechenplans nicht lohnt oder über die Tastatur ein neuer Rechenplan erstellt werden soll. Der Lochstreifen kann bis zu zwölfstellige Befehle abarbeiten, wobei pro Zeile maximal acht Lochungen möglich sind. Die weiteren vier werden in der nächsten Zeile gelocht. In Vorbereitung auf den Einsatz von bis zu 1024 Speicherzellen sind die 2^{12} Befehle notwendig. Zwei Bit werden für die Art der Speicher- oder Ablese-Operation benötigt, die weiteren zehn Bit zum Adressieren der Speicherzellen.

Im Allgemeinen wird, wie bei der Z3, mit achtstelligen Befehlen gearbeitet. In der ersten Ausbaustufe wird ein Speicher mit 64 Zellen verwendet. Daher ist ein Umschalten auf zwölfstellige Befehle nicht notwendig. Die Nutzung größerer Speicher erfolgte erst an der ETH in der Schweiz. Für die Verwendung des größeren Speichers muss der Z4 bei jedem Speicher- oder Ablesebefehl zuerst das Anzeigen von zwölfstelligen Befehlen kommandiert werden. Die Z4 muss jetzt eine weitere Zeile im Lochstreifen lesen, um die vier weiteren Bit auszulesen. Nachdem der Speicher mit zwölfstelligen Befehlen angesprochen wurde, ist der Rückstellbefehl notwendig. Dieser bringt die Z4 in den Grundzustand und es werden wieder achtstellige Befehle

erwartet und verarbeitet. Da außer den Ablese- und Speicherbefehlen alle anderen Befehle achtstellig codiert sind.

Das Lampenfeld für die Interaktion mit dem Bediener ist eine komplette Neuerung in der Z4 und erleichtert die Eingabe von Variablen oder Startwerten. Die Lampenfeldbefehle, welche die Lampen auf dem Lampenfeld steuern, ermöglichen so die Interaktion mit dem Programmierer. Eine detaillierte Analyse des Lampenfelds wird in Abschnitt 5.6 gegeben.

Von den eigentlich geplanten 32 Rechenoperationen sind nur 24 realisiert worden. Die Zuordnung, welche Rechenoperation ausgeführt werden soll, wird über die sechs niederwertigsten Bits codiert und auf die Pb_{0-5} Relais gegeben. Die beiden höherwertigsten Bit codieren die $Pb_{10,11}$ Relais und legen damit die Art der Operation fest. $\overline{Pb_{10}} \wedge \overline{Pb_{11}}$ zeigen Steuerbefehle an. Ein Ansprechen des Speichers wird durch Pb_{11} angezeigt. Gilt $Pb_{11} \wedge \overline{Pb_{10}}$, handelt es sich um einen Speicherbefehl. Bei $Pb_{11} \wedge Pb_{10}$ um einen Ablesebefehl. $\overline{Pb_{11}} \wedge Pb_{10}$ signalisiert eine Rechenoperation. Eine weitere Unterscheidung der Operationen erfolgt durch Pb_{0-5} .

Erfolgt die Eingabe nicht über einen Rechenplan, sondern über die angeschlossene Tastatur, wird jeder getasteten Operation eine eindeutige Kombination von einem Pn_{0-7} und einem Po_{10-14} Relais zugeordnet. Dieses belegen dann die Pb Relais entsprechend der Codierung. Eine ausführliche Beschreibung dieser Zuordnung erfolgt in Kapitel [5.3.1].

Eine Übersicht über die Rechenoperationen und deren Codierung zeigt folgende Tabelle:

Klasse	Beschreibung	Befehl	Po	Pn	Befehlscode
Steuerbefehl	Start	St	14	0	00 100000
	Schluss	Sh	14	1	00 100001
	Umstellen	Pu_1	16	4	00 11-100
	Rückstellen	Pu_3	16	5	00 11-101
Lampenfeld	x-Richtung	Lx	16	0	00 11-000
	Heimlauf x	Hx	16	1	00 11-001
	y-Richtung	Ly	16	2	00 11 010
	Heimlauf y	Hy	16	3	00 11-011
Speicherbefehle	Speichern	Ps z	20		10 $Z_6 Z_5 Z_4 Z_3 Z_2 Z_1 Z_0$
	Ablesen	Pa z	21		11 $Z_6 Z_5 Z_4 Z_3 Z_2 Z_1 Z_0$
Ein-/Ausgabe	Eingabe übertragen	Lu	12	6	01 011110
	Ergebniss anzeigen	Ld	12	7	01 011111
Arithmetik	-a-b	Ls	10	0	01 000000
	b-a	Ls	10	1	01 000001
	a-b	Ls	10	2	01 000010
	a+b	Ls	10	3	01 000011
	a*b	Lmo	10	4	01 000100
	a/b	Li	10	5	01 000101
	Maj	Lj	10	6	01 000110
	Min	Lj	10	7	01 000111
	Sign	Lg	11	0	01 001000
	Fpos	Lp	11	1	01 001001
	x(-1)	Lh_1	11	2	01 001010
	$ x $	Lh_1	11	3	01 001011
	x^2	Lh_5	11	4	01 001100
	$1/x$	Lh_5	11	5	01 001101
	$\sqrt[3]{x}$	Lw_1	11	6	01 001110
	$\sqrt[3]{x}$	Lw_1	11	7	01 001111
	$x * 1/2$	Lh_2	12	0	01 011000
	$x * 2$	Lh_2	12	1	01 011001
	$x * 1/10$	Lh_3	12	2	01 011010
	$x * 10$	Lh_3	12	3	01 011011
	$x * \Pi$	Lm_1	12	4	01 011100
	$x * g$		12	5	01 011101

5 Blockarchitektur der Z4

In diesem Kapitel, werden die einzelnen Komponenten der Z4 detaillierter betrachtet und unter dem Gesichtspunkt aktueller Blockarchitekturen analysiert. Hierunter fallen das Rechenwerk mit integriertem „Prozessor“, das Planwerk welches als Steuereinheit die Abläufe und Koordination der internen Prozesse übernimmt, die Ein- und Ausgabe von Dezimalzahlen und die dazu entsprechende Umwandlung ins Dualsystem, das neuartige Lampenfeld, welches die Eingabe und Ausgabe von Variablen erleichtert und zum Schluss noch die Architektur des Speichers.

5.1 Rechenwerk

Das Rechenwerk führt die einzelnen Rechenoperationen durch und wird durch das Leitwerk gesteuert. Es ist ein Zusammenschluss von Registern, welche untereinander, mit der Tastatur, dem Speicher und einigen Kontrolllampen verbunden sind. Das Herz sind die zwei Additionswerke A und B. Im Additionswerk A werden die Exponenten und in Additionswerk B die Mantissen addiert bzw. subtrahiert. Die Ein- und Ausgabe erfolgt über die Ta Relais der Tastatur. Der Speicher kann Ergebnisse aufnehmen oder Operanden zur Verfügung stellen.

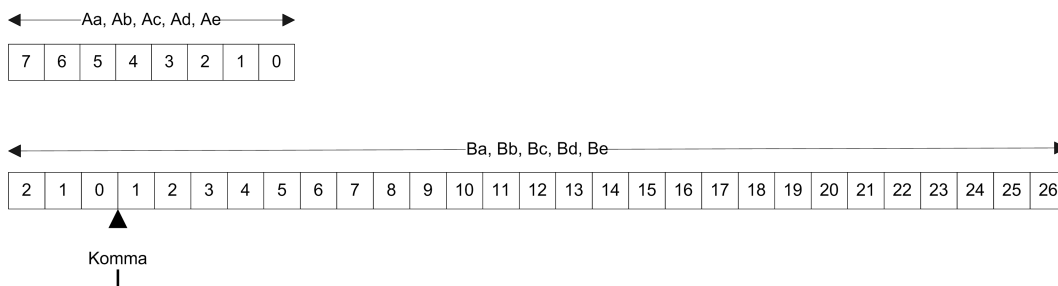


Abbildung 3: Bit Aufteilung der Register im Rechenwerk.

Die Register im Additionswerk A haben alle eine Länge von acht Bit. Im Speicherwerk werden für den Exponenten allerdings nur sieben Bit gespeichert. Das zusätzliche Bit erhöht die Genauigkeit während der Rechnung und verringert die Gefahr eines Über- oder Unterlaufs. Im Additionswerk B sind die Register Bf, Bg mit 24 Stellen für die Mantisse vorgesehen. Die Register Ba, Bb und Be besitzen 29 Bit, von denen 26 Bit für die Nachkommastellen und bis zu drei Stellen vor dem Komma möglich sind. Gespeichert werden können auch hier nur die 22 Bit hinter

dem Komma. Die zusätzlichen Bit erhöhen die Genauigkeit bei den Rechnungen und verringern Rundungsfehler.

Die Additionswerke A und B sind über die Tastaturrelais Ta mit der Ein- und Ausgabe verbunden, sowie mit dem Speicher C. Gesteuert wird der Zahlenkreislauf durch die Steuerrelais E und F, welche durch das Öffnen und Schließen die Übertragung von einem Register auf ein anderes ermöglichen. Die Register des Additionswerks A werden durch die Steuerrelais E und die des Additionswerks B durch die Steuerrelais F gesteuert. Von der Tastatur kommende Werte werden entsprechend dem Übersetzungsalgorithmus auf Af und Bf gegeben.

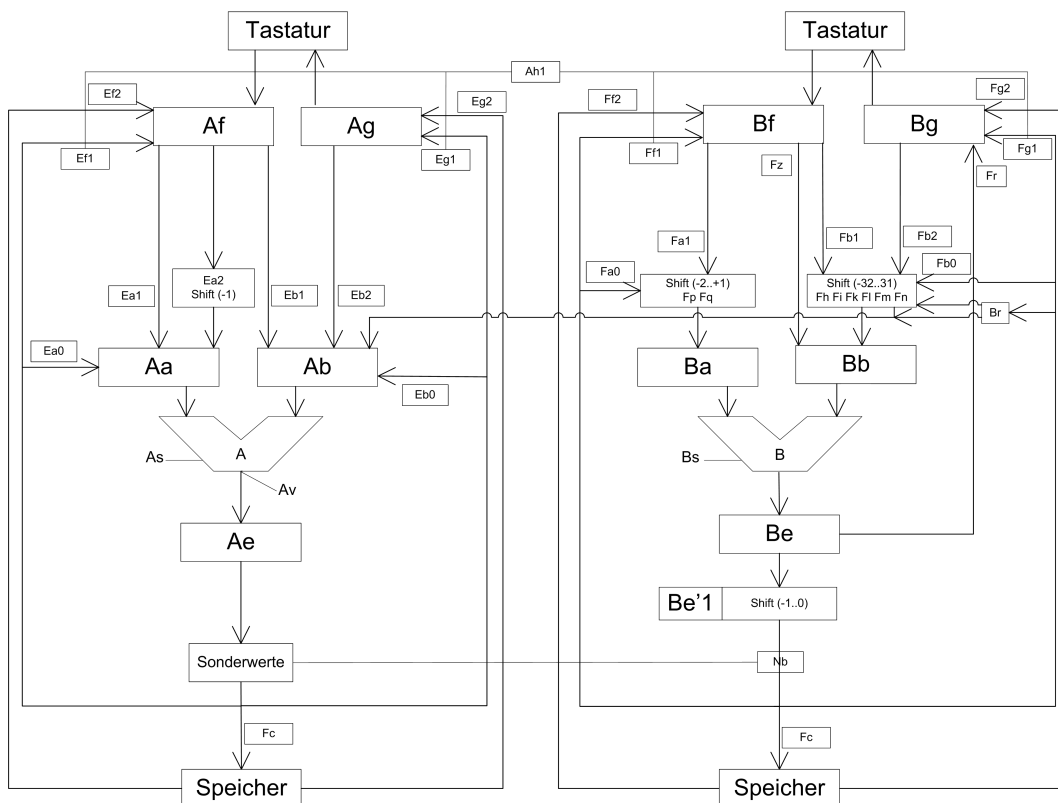


Abbildung 4: Rechenwerk der Z4

Von den Tastaturrelais wird die Anzeige von Rechenergebnissen auf dem Lampenfeld gesteuert. Eingaben im Dezimalsystem werden durch eine automatisch stattfindende Umformung ins Dualsystem übersetzt und auf die Af, Bf Relais des Rechenwerks gegeben. Von dort werden die Werte in Vierergruppen für die Übersetzung ins Dualsystem von Bf durch Fz an Bb übertragen. Zwischen- oder Endergebnisse, welche

ausgegeben werden sollen, werden von Be wiederum in Vierergruppen durch Fr auf die Bg Relais gegeben, um von dort auf die Ta Relais der Tastatur übertragen zu werden.

Der Exponent des ersten Operanden wird von den Tastaturrelais auf Af, der Exponent des zweiten Operanden auf das Register Ag gegeben. Die Mantisse entsprechend auf Bf bzw. Bg für den 2. Operanden. Die Übertragung von Ergebnissen auf die Ta Relais zur Ausgabe erfolgt nur über die Ag und Bg Relais. Das Ergebnis einer Rechenoperation wird immer über Ae und Be ausgegeben und von dort über den Zahlenkreislauf auf die anderen Register geleitet. Das Ergebnis einer Operation kann von Ae, Be entweder über Fc auf den Speicher übertragen werden oder in den Zahlenkreislauf gegeben werden.

Soll ein Wert vom Speicher auf die Additionswerke gegeben werden, geschieht dieses durch Ef_2 für Alpha auf Af oder durch Eg_2 für Alpha auf Ag. Die Übertragung des Exponenten auf Ag erfolgt nur, wenn das Register Af bereits belegt ist. Die Übertragung der Mantisse erfolgt durch Ff_2 auf Bf oder durch Fg_2 auf Bg wenn Bf schon besetzt ist. Das Additionswerke A und B sind ähnlich aufgebaut.

In Additionswerk A kann der erste Operand über die Steuerung von Ea_2 auf Aa gegeben werden und wird gleichzeitig um eine Stelle nach unten verschoben. Das entspricht der Division durch zwei. Ea_0 veranlasst einen blinden Umlauf und gibt das Ergebnis der soeben durchgeführten Rechnung wieder auf Aa. In Additionswerk B können Werte, die auf das Register Ba geleitet werden, über die kleine Weichenstrasse von $(-2..+1)$ verschoben werden. Die Steuerung erfolgt durch die Stellung der Fp und Fq Relais. Werte, die auf das Register Bb geleitet werden, können über die große Weichenstrasse um bis zu $(-32..+31)$ verschoben werden, wobei eine Verschiebung nur um bis zu 24 Stellen Sinn macht, da es sonst zu einem Unterlauf kommen kann. Die Steuerung der Verschiebung erfolgt durch die Fh..Fn Relais

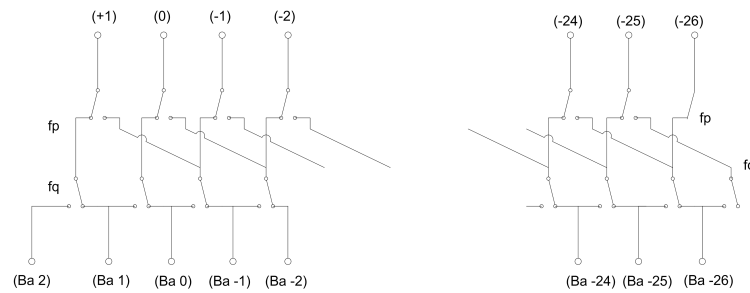


Abbildung 5: Kleine Weichenstrasse

Handelt es sich um eine Subtraktion oder die Vorzeichen beider Operanden sind bei einer Addition unterschiedlich, wird Bt geschaltet, welches das Supplement des negativen Operanden bildet. Damit müssen die Additionswerke nur eine einfache Addition durchführen. Das Vorzeichen, des Ergebnisses einer Addition wird durch das Relais Av signalisiert.

Durch Ah_1 kann am Ende einer Operation, das löschen der Operandenregister Af, Bf und Ag, Bg verhindert werden. Die genaue Funktionsweise wird im folgenden Kapitel u.a. in Abbildung 8 veranschaulicht. Bei einer Addition kann die Mantisse den Wert ≥ 2 annehmen und muss dann normalisiert werden. Dafür wird die Mantisse über $Be'1$ um eine Stelle nach unten verschoben. Der Exponent muss daraufhin um eins erhöht werden. Bei einzelnen Rechnungen kann die Mantisse auch den Wert < 1 annehmen und muss über das Relais Br ausgerichtet werden, damit vor dem Komma eine eins steht. Gleichzeitig muss der Exponent um die entsprechende Anzahl an Verschiebungen verkleinert werden.

Über die Schaltung N, dargestellt durch das Relais Nb, wird überprüft, ob es sich bei dem Ergebnis um ein gültiges handelt. Über die Nb Relais wird angezeigt, ob das Ergebnis sehr klein, sehr groß, unendlich, 0 oder unbestimmt ist. Daraufhin werden die Register des Exponenten wie in Abbildung 23 belegt.

5.2 Leitwerk

Das Leitwerk steuert die einzelnen Zahlenübertragungen im Rechenwerk. Durch eine geschickte Aktivierung der verschiedenen E und F Relais, sowie weiterer Steuerrelais können alle Rechenoperationen gesteuert werden. Dazu werden die verschiedenen Steuerrelais aktiviert und ermöglichen damit den Inhalt eines Registers auf ein anderes zu übertragen.

Bei Übersetzen einer Zahl aus dem Dezimalsystem ins Dualsystem entfällt die folgende Schaltung. Es muss kein Operand belegt sein, sondern nur von der Tastatur das Fertigsignal der Eingabe durch Tf_4 signalisiert werden. Durch den Befehl „Übersetzen“ erfolgt diese, gesteuert durch eine Relaiskette.

Alle anderen Rechenoperationen werden über die Bedingung $\overline{Pb_{11}} \wedge Pb_{10}$ angezeigt. Diese werden wiederum in Operationen mit einem und mit zwei Operanden unterschieden. Rechenoperationen mit einem Operand werden durch $Pb_3 \vee Pb_4$ angezeigt und können gestartet werden, sobald $Af_{10} \wedge \overline{Pb_{11}} \wedge (Pb_3 \vee Pb_4)$ gilt. $\overline{Pb_3} \wedge \overline{Pb_4}$ signalisiert eine Rechenoperation mit zwei Operanden.

Der erste Operand wird auf die Register Af, Bf und der zweite auf die Register Ag, Bg gegeben. Die Aufteilung der jeweiligen Operanden erfolgt entsprechend in Exponent A und Mantisse B. Die Übertragung auf die Operandenrelais kann vom Speicher, der Tastatur oder aus dem Zahlenkreislauf erfolgen.

Dass der erste Operand belegt ist, wird durch Af_{10} signalisiert, dass der zweite Operand belegt ist durch Ag_{10} . Wird der zweite Operand vom Speicherwerk eingelesen, wird dieses durch Cr_2 angezeigt und ersetzt damit die Funktion von Ag_{10} .

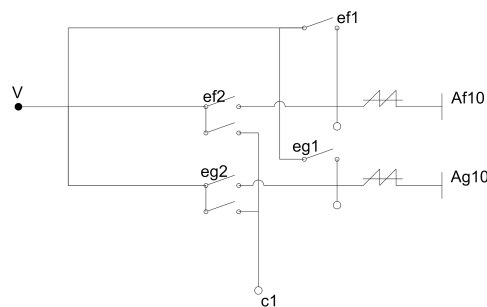


Abbildung 6: Schaltung der Besetztrelais der Operandenregister

Die Übertragung des Exponenten vom Speicher auf Af erfolgt über die Aktivierung von Ef_2 , auf Ag durch Eg_2 und damit auf den Platz für den zweiten Operanden. Das Füllen der Exponentenregister aus dem Zahlenkreislauf ist über Ef_1 und Eg_1 möglich. Wird Ef_1 oder Ef_2 aktiviert um einen Operanden auf das Register Af zu legen, wird gleichzeitig Af_{10} aktiviert. Ag_{10} bei einer Belegung des Ag Registers.

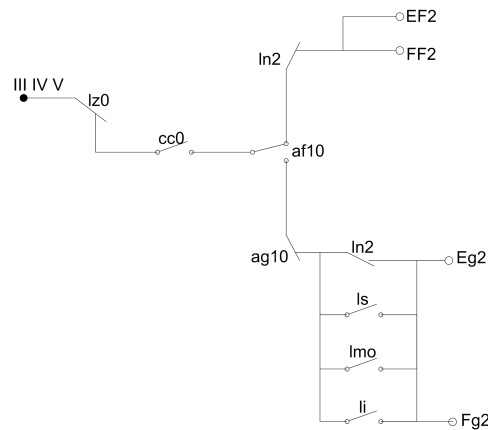


Abbildung 7: Übertragung des 1. und 2. Operanden auf die Register im Rechenwerk

Die Übertragung einer Zahl vom Speicher auf die Operandenregister wird durch Ef_2 und Ff_2 auf die Af, Bf Relais, also die Register des ersten Operanden gesteuert. Voraussetzung ist, dass diese nicht belegt sind, was durch $\overline{Af_{10}}$ angezeigt wird. Ebenfalls muss auf den Cc Relais des Speichers bereits ein Wert in Bereitschaft stehen und darauf warten, auf das Rechenwerk übertragen zu werden. Dies wird durch ein aktiviertes Cc_0 Relais signalisiert. Weitere Voraussetzung ist, dass das Relais lz_0 nicht aktiviert ist. Wird das Ergebnis der gerade abgeschlossenen Rechenoperation auf den Speicher übertragen, werden gleichzeitig die Operandenrelais Af, Bf, Ag und Bg gelöscht und stehen für die Aufnahme neuer Operanden zur Verfügung.

Sind die Register Af, Bf des ersten Operanden bereits belegt, durch Af_{10} angezeigt, kann Eg_2 und Fg_2 aktiviert werden. Dadurch erfolgt die Übertragung des Speicherwertes auf die Ag, Bg Relais. Voraussetzung ist hier ebenfalls die gleiche Stellung der Cc_0 und Lz_0 Relais, sowie das durch $\overline{Ag_{10}}$ angezeigt wird, dass die Register Ag und Bg nicht belegt sind. Findet gerade keine Operation statt, durch ln_2 angezeigt, oder ist eine Operation mit zwei Operanden ls , lmo , li kommandiert, kann der Wert auf die Ag und Bg Relais übertragen werden.

Das Ergebnis einer Rechnung wird von den Additionswerken auf die Register Ae und Be gegeben. Ist eine Rechnung beendet muss der Wert von Ae und Be auf ein anderes Registerpaar übertragen werden, damit eine weitere Rechnung stattfinden kann und das Ergebnis der vorherigen nicht überschrieben oder gelöscht wird. Die Übertragung von Ae und Be erfolgt durch Ef_1 , Ff_1 oder Eg_1 , Fg_1 abhängig davon auf welches Register der Wert gelegt werden soll. Das Ergebnis bleibt so im Rechenwerk und wird auf eines der Operandenregister übertragen. Alternativ kann das

Ergebnis auch in den Speicher geschrieben werden. Zu diesem Zeitpunkt liegt schon der nächste Befehl auf den Pb Relais. Handelt es sich dabei um einen Speicherbefehl, wird über Lo_0 , Lo_1 und darüber Lc geschaltet. Das Ergebnis muss nun in den Speicher übertragen werden. Dafür müssen dessen Aufnahmeregister Ca bereit sein. Wird durch $\overline{Ca_0}$ angezeigt, dass sich kein Wert auf den Ca Registern befindet, kann Fc aktiviert werden. Befindet sich auf den Ca Relais noch ein Wert, muss abgewartet werden, bis dieser vollständig auf das Speicherwerk übertragen ist. Es erfolgt ein blinder Umlauf durch das Additionswerk durch Ea_0 und Fa_0 gesteuert. Im nächsten Spiel wird geprüft ob der Speicher zur Verfügung steht. Ist das Register Ca frei, wird Fc aktiviert, andernfalls erfolgt erneut ein blinder Umlauf.

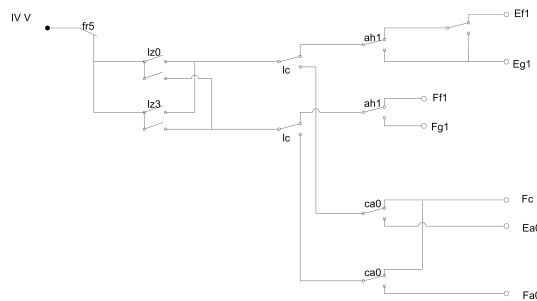


Abbildung 8: Übertragung des Ergebnisses auf weitere Register

Tritt der Fall ein, dass das Ergebnis im Rechenwerk gehalten und auf die Register des ersten Operanden gegeben werden soll, muss $\overline{Lc} \wedge \overline{Ah_1}$ gelten. Der Wert des ersten Operanden wird damit durch das Ergebnis der Rechenoperation überschrieben. Tritt der Fall ein, dass eine Rechnung die wiederholte Multiplikation mit einem Operanden zur Folge hätte, wie es x^3 der Fall ist, verhindert Zuse mit seiner Schaltung die Notwendigkeit Operanden erneut auf die Operandenregister einzulesen. Bei der Multiplikation wird im Operationsbefehl zur Codierung der Multiplikation zusätzlich Pb_5 aktiviert und somit Ah_1 aktiviert. Das Ergebnis der ersten Rechnung ist x^2 und wird auf die Operandenregister Ag, Bg geleitet. Eine erneute Multiplikation mit dem ersten Operanden und dem zweiten, welcher schon das Quadrat des ersten Operanden beinhaltet, führt zu einem schnellen Ergebnis. Hierfür gilt $\overline{Lc} \wedge Ah_1$. In Abhängigkeit auf welches Register das Ergebnis von Be geleitet wird, wird gleichzeitig das passende Relais Af_{10} bzw. Ag_{10} aktiviert.

Voraussetzung für die Übertragung des Ergebnisses auf ein anderes Register nach Ende der Rechenoperation ist die Bedingung Lz_0 . Lz_0 wird, wie in der Analyse der

arithmetischen Algorithmen noch gezeigt wird, im letzten Spiel jeder Rechenoperation aktiviert. Das Ende der Rechenoperation ist somit erfolgt und die Übertragung des Ergebnisses ist wie beschrieben möglich. Durch das gleichzeitige lösen der Selbsthaltekreise der Operationsbefehle, kann eine neue Rechenoperation kommandiert werden. Zusätzlich wird über Lz_0 das Folgerelais von Lz_1 geschaltet, welches Ah_1 wieder deaktiviert.

Findet ein blinder Umlauf statt und die Übertragung auf das Speicherwerk steht noch aus, wird dieses durch Lz_3 angezeigt. Für alle vier geschilderten Fälle ist ein deaktiviertes Fr_5 Relais Voraussetzung. Erfolgt die Operation „Rückübersetzen“, welche das Ergebnis der Rechnung auf dem Lampenfeld zur Anzeige bringt, muss das Ergebnis der Rechnung aus dem Dualsystem zurück ins Dezimalsystem übersetzt werden. Die Übertragung von Be auf das Operandenregister entfällt damit und über Dz_2 werden die Relais Ag , Bg gelöscht. Dieses wird mit einem aktiviertem Ruhekontaktrelais Fr_5 erreicht.

5.3 Planwerk

Die Steuerung der Z4 erfolgt über die Schaltungen des Planwerks. Es übernimmt die Koordinierung und Steuerung der Ein-/Ausgabe, der Weiterleitung der Operations- und Speicherbefehle an das Rechenwerk und den Speicher.

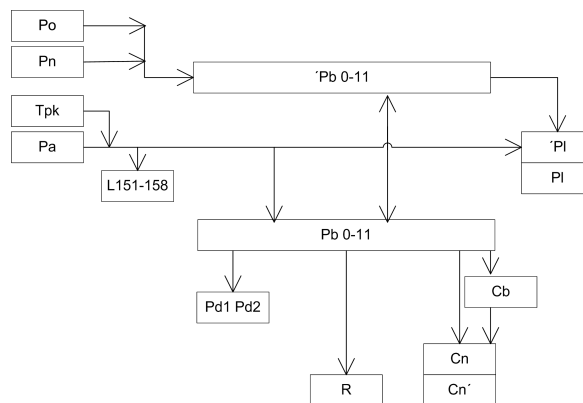


Abbildung 9: Befehlskreislauf im Planwerk

Die Abbildung zeigt die verbundenen Komponenten, welche durch die Schaltungen des Planwerks kontrolliert und gesteuert werden. Das Herz sind die Relais Pb_{0-11} , welche mit den $'Pb_{0-11}$ parallel geschaltet werden können und die Befehle zur Steuerung der Z4 halten. Die bis zu zwölfstelligen Befehle werden durch die Relais des

Abtasters Pa oder die dem Abtaster parallel geschalteten Kontrolltasten Tpk auf die Pb Relais gegeben. Die Lampen L151-L158 können zu Kontrollzwecken die richtige Funktionsweise des Abtasters anzeigen.

Die auf den Pb Relais befindlichen Befehle, können die Schrittschalter Pd_1 und Pd_2 steuern, um so den Zeiger auf dem benutzergesteuerten Lampenfeld auf eine andere Position zu bewegen. Operanden werden auf das Rechenwerk R geleitet. Zur Anwahl der richtigen Speicheradresse wird diese über Cn auf das Wählwerk des Speichers gegeben. Steht der Speicher nicht zur Verfügung, kann die Adressierung der Speicherzelle auf Cb und später von Cb auf Cn gegeben werden.

Eingaben von der Tastatur werden über die Kombination von jeweils einem Po und Pn Relais eindeutig codiert und belegen die $'Pb$ Relais entsprechend der achtstelligen Codierung. Sind die $'Pb$ und Pb Relais getrennt voneinander geschaltet, können die Befehle der Tastatureingabe direkt auf die Relais $'Pl$ gegeben werden, welche die Lochung über Pl aktivieren. Durch die Trennung von $'Pb$ und Pb kann ein Rechenplan im laufenden Betrieb erstellt werden, ohne die laufenden Rechnungen der Maschine zu stören.

Soll die Steuerung der Maschine über die direkte Eingabe der Tastatur erfolgen, müssen die Pb und $'Pb$ Relais parallel geschaltet werden, um die Befehle von den $'Pb$ Relais auf die parallel geschalteten Pb und dann auf das Rechenwerk, Speicher oder Lampenfeld weiter zu leiten. Soll ein Rechenplan vom Abtaster eingelesen und kopiert werden, müssen die eingelesenen Befehle direkt auf den Locher Pl gegeben werden. Hierfür werden die Pb und $'Pb$ Relais nicht gebraucht. Anders sieht es aus, wenn der Rechenplan durch Tastatureingaben erstellt werden soll. Die Tastatureingaben landen auf den $'Pb$ und werden von dort an die $'Pl$ Relais des Lochers weiter geleitet.

Das Weiterleiten der Befehle zwischen den unterschiedlichen Registern des Planwerk erfolgt durch PP_{1-7} , auf die noch näher eingegangen wird.

5.3.1 Tastatur - Planwerk

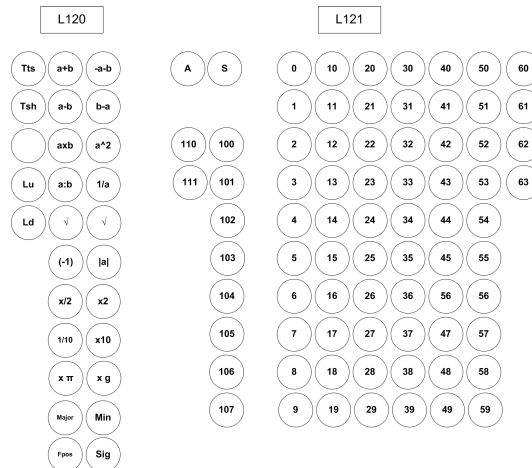


Abbildung 10: Operations- und Speichertastatur

Bei der Eingabe von Befehlen über die Tastatur werden diese auf die $'Pb$ Relais gebracht. Die Abbildung zeigt die Aufteilung der Tasten auf der Tastatur. Es wird in Operations-, Speicher- und Ablesebefehle unterschieden. Links liegen die Operationstasten, welche die Relais Tpo_{0-31} ansprechen und damit eine Rechenoperation kommandieren. Rechts liegen die Speicherstellentasten Tpn . Die in der Mitte liegenden Tasten Tpa und Tps kommandieren das Auslesen bzw. das Abspeichern in eine Speicherzelle. Es können über diese Tastatur bis zu 1024 Speicherzellen angesprochen werden. Die Aufteilung erfolgt auf zwei Speicherwerke, die jeweils mit den Tasten Tpn_{110} oder Tpn_{111} ausgewählt werden können. Jedes Speicherwerk besteht aus acht Zellengruppen mit jeweils 64 Speicherzellen. Die gewünschte Zellengruppe kann durch die Tasten $Tpn_{100-107}$ gewählt werden. Die einzelne Zelle in der Zellengruppe wird über das direkte Drücken einer Taste Tpn_{0-63} adressiert. Die über den Tasten angebrachten Lampen L120 und L121 zeigen an, in welchem Moment ein Operationsbefehl L120 oder ein Speicherzellenbefehl L121 zu kommandieren ist. Sollte durch L120 die Eingabe eines Operationsbefehls angezeigt werden, ist das Drücken der Speichertasten ohne Wirkung. Wird ein Speicherbefehl erwartet, bleibt das Drücken der Operationstasten ohne Wirkung.

Der allgemeine Befehlssatz wurde bereits in der Übersicht der Z4 vorgestellt. In Zusammenhang mit dem Planwerk soll im Folgenden ein detaillierter Überblick über die Codierung der Befehle in Bezug auf das Planwerk gegeben werden.

Die Operationsbefehle sind als achtstellige Befehle codiert und werden beim Einlesen

auf die Relais Pb_{0-5} und Pb_{10-11} gegeben. Handelt es sich um zwölfstellige Speicher- oder Ablese-Befehle kommen noch die Relais Pb_{6-9} dazu. Werden die Befehle vom Lochstreifen abgelesen, wird an den Stellen mit einer Lochung im Lochstreifen, das entsprechende Pb Relais über den Abtaster als aktiv gesetzt.

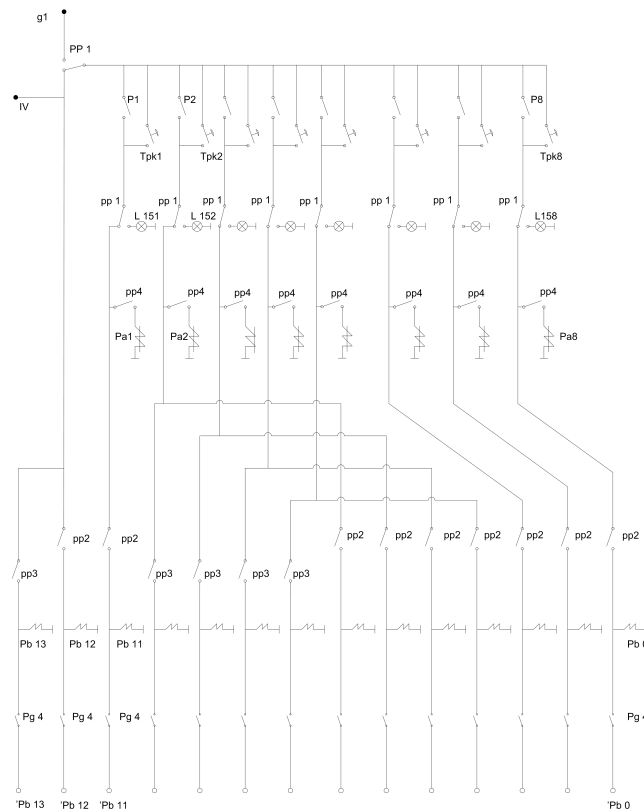


Abbildung 11: Schaltung des Befehlskreislauf im Planwerk

Aus der Schaltung wird ersichtlich, dass die Eingaben über den Abtaster den Relais P_{1-8} zugeordnet werden. Die den Relais P_{1-8} parallel geschalteten Schalter Tpk_{1-8} können die Funktion des Abtasters überbrücken und so die richtige Funktionsweise der Schaltung überprüfen. Das Aktivieren von PP4 leitet den eingelesenen Befehl auf Pa_{1-8} von wo sie über PP7 auf die 'Pl Relais des Lochers weitergeleitet werden können. Eine Kopie eines Rechenplans lässt sich so einfach erzeugen. In der von Zuse eingereichten Patentanmeldung hat die Schaltung den Aufbau wie in der Abbildung. PP2 leitet den vom Abtaster eingelesenen Wert auf die Pb_{0-6} und Pb_{11-12} Relais. PP3 bringt bei zwölfstelligen Befehlen die zusätzlichen vier Bit auf die Relais Pb_{7-10} , sowie Pb_{13} . Es handelt sich hierbei um einen Fehler in der Konstruktionszeichnung von Zuse. Die vier zusätzlichen Bit, müssten sich auf den Relais Pb_{6-9} und nicht Pb_{7-10} befinden. Das Relais Pb_{12} zeigt das Vorhandensein eines achtstelligen Befehls

auf den Pb Relais und Pb_{13} das eines zwölfstelligen an. Sind die Pb und 'Pb Relais durch Pg4 parallel geschaltet, wird der Wert vom Abtaster gleichzeitig auf die Pb und ' Pb Relais gegeben.

Werden die Befehle über die Tastatur eingegeben, erfolgt die Codierung der einzelnen Operationsbefehle während der Eingabe durch die Pn und Po Relais. Die Zuordnung der Po und Pn Relais zu den einzelnen Befehlen wurde bereits im Kapitel 4.4 gegeben. Die Pn und Po Relais veranlassen dann wiederum die richtige Belegung der ' Pb Relais.

Wird z.B. der Befehl „Multiplikation“ gegeben, wird dies durch drücken der Tpo_4 Taste erreicht. Dadurch werden die Relais Po_{10} und Pn_4 aktiviert. Die führt durch die Schaltung [A.3.1] zur Belegung der ' Pb_2 und ' Pb_{10} Relais. Dieses entspricht somit wieder der Codierung, wie sie über den Lochstreifen erfolgt wäre.

Das Ansprechen der acht verschiedenen Pn_{0-7} Relais wird durch eine Codierung der ' Pb_{0-2} Relais dargestellt. Die Operationsbefehle sind in Gruppen zu acht Befehlen zusammengefasst und belegen die ' Pb_{0-2} Relais eindeutig. Jede Gruppe belegt ein eindeutiges Po Relais von Po_{10-13} , welche die Relais ' Pb_{3-4} entsprechend belegt. ' Pb_{6-9} steht für die zwölfstelligen Speicherbefehle zur Verfügung, wenn kein Operationsbefehl vorliegt. ' Pb_{10-11} zeigt die Art des Befehls an und unterscheidet nach Belegung in Rechenoperation, Speicher-, Ablesebefehl oder allgemeine Steuerung. Jeder über die Tastatur kommandierte Befehl, belegt eine eindeutige Kombination aus Po und Pn Relais. Die Belegung der Po Relais wird in Abbildung 12, die der Pn in Abbildung 13 gezeigt.

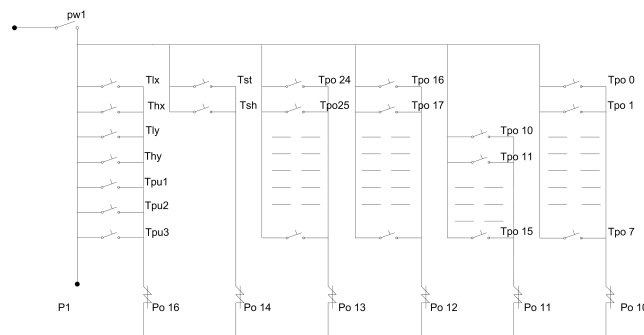


Abbildung 12: Schaltung Po Relais durch Tastatureingabe

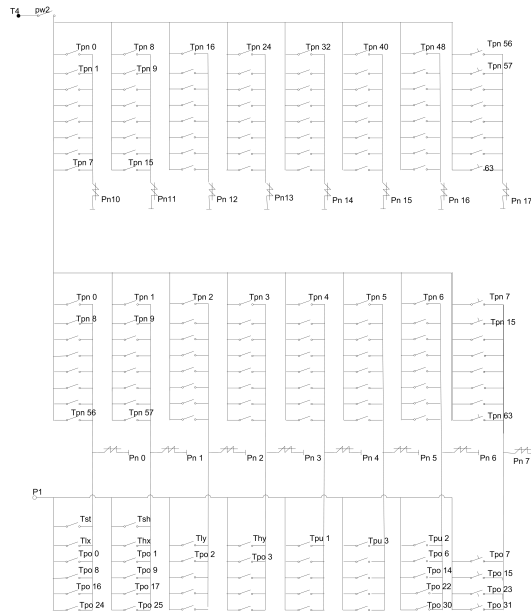


Abbildung 13: Schaltung der Pn Relais durch Tastatureingabe

Wird über die Tastatur ein Speicher oder Ablesebefehl gegeben, muss Tps oder Tpa gedrückt werden, welches dann das Relais Po_{20} bzw. Po_{21} aktiviert. Ist eines der beiden Relais aktiviert wird hierdurch Pw_2 aktiviert, welches den Wechsel der Lampenanzzeige von L120 auf L121 veranlasst. Dieses setzt allerdings die Aktivierung von $'Pb_{12}$ voraus, was die Belegung der $'Pb$ Relais mit einem mindestens achtstelligen Wert bedeutet. Sobald ein Wert vom oder zum Speicher übertragen wurde, wird über Pn_{52} der Selbsthaltekreis von Po_{20} bzw. Po_{21} gelöscht. Es wird somit wieder die Lampe L120 aktiviert. Als nächster Befehl wird die Tastung einer Rechenoperation erwartet.

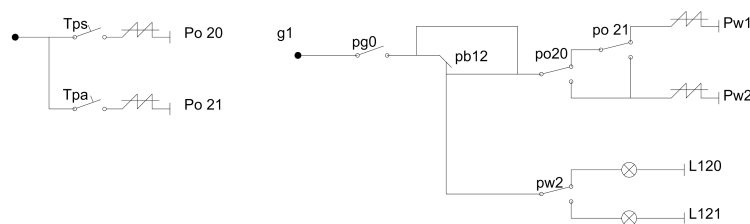


Abbildung 14: Wechsel zwischen Operations- und Speicherbefehl

Ohne die Aktivierung von Po_{20} oder Po_{21} bleibt Pw_1 aktiviert und ermöglicht die Eingabe eines Operationsbefehls. Das Drücken einer Operationstaste, bei aktiviertem Pw_1 Relais führt automatisch zur Aktivierung eines der Po_{10-16} Relais und in einer weiteren Schaltung zur passenden Aktivierung eines der Pn_{0-7} Relais.

Handelt es sich um einen Speicher oder Ablesebefehl und Po_{20} oder Po_{21} ist aktiviert, wird dadurch Pw_1 ab und Pw_2 aktiv geschaltet. Die Anzeige der Lampe L120 wechselt auf L121 und das Drücken einer Operationstaste würde keine Auswirkungen haben. Durch die Aktivierung von Pw_2 wird durch Drücken einer Speicherzellentaste Tpn, jeweils ein Relais der Gruppen Pn_{10-17} und Pn_{0-7} aktiviert. Jeder Speicher und Operationsbefehl wird durch eine eindeutige Kombination der Relaisgruppen Pn und Po codiert. Das Drücken einer weiteren Operationstaste, würde ein weiteres Relais aktivieren. Eine eindeutige Codierung der 'Pb Relais ist damit nicht mehr möglich. Dass nur ein Relais der Gruppe Po_{1-7} , Pn_{10-17} und Po_{10-16} aktiviert ist, wird durch eine Prüfschaltung kontrolliert. Diese schaltet die Relais, Pq_3 , Pq_2 und Pq_1 .

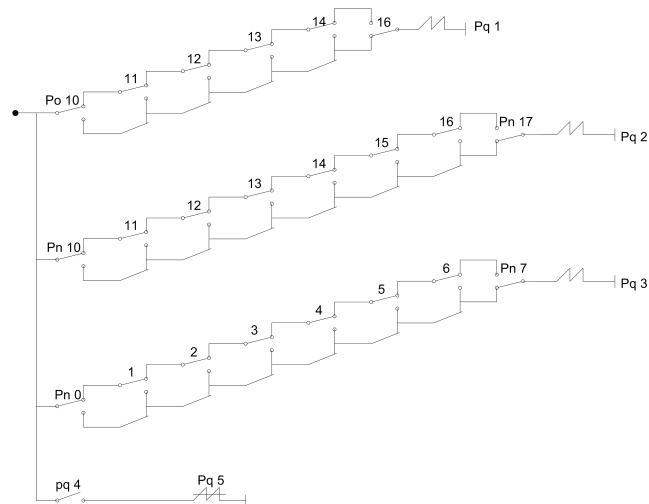


Abbildung 15: Schaltung zur Prüfung auf eindeutige Tastatureingabe

Handelt es sich um einen Speicherbefehl, müssen bei eindeutiger Tastung die Relais Pq_1 und Pq_2 ansprechen, welche das Relais Po_{52} aktivieren. Po_{52} löst den Selbsthaltungskreis der Po_{20} und Po_{21} Relais, dass wieder die Lampe L120 für die Operationseingabe leuchtet und Pw_1 aktiviert ist. Gleichzeitig wird durch Po_{52} die Schaltung der passenden 'Pb_{0-5,10,12} Relais veranlasst. Durch die Schaltung von Pn_{52} werden die 'Pb₀₋₁₂ mit dem eingetasteten Operationsbefehl codiert. Die eingetasteten Befehle befinden sich nun auf den 'Pb Relais. Die Schaltung „Belegung der Pb Relais durch Tastatureingabe“ befindet sich im Anhang [A.3.1].

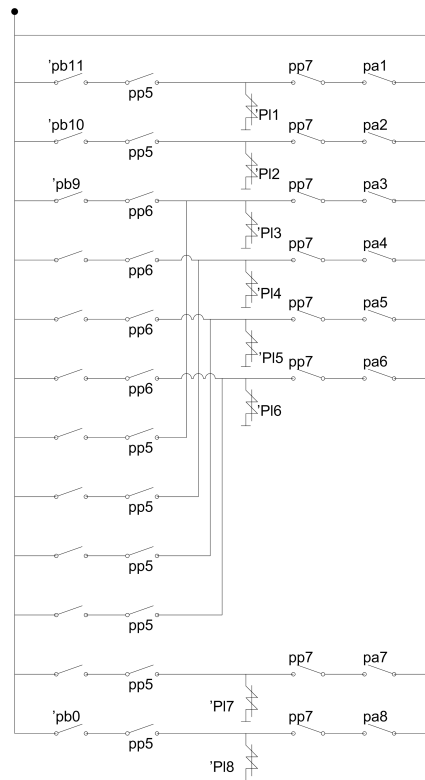


Abbildung 17: Übertragung der Befehle auf den Locher

Die Erstellung eines Rechenplans ist so über die Eingabe der Tastatur und die 'Pb Relais möglich. Gleichfalls ist auch das Erstellen einer Kopie eines bestehenden Rechenplans über die Pa Relais, über den Abtaster vorgesehen. Die vom Abtaster gelesenen Befehle können über PP4 auf die Pa Relais übertragen werden, um eine Lochung des Befehls durch die PP7 zu veranlassen. Die Werte vom Abtaster können bei achtstelligen Befehlen über PP2 und bei zwölfstelligen über PP2 und PP3 gesteuert, auf die Pb Relais übertragen werden. Sind die *Pb* und '*Pb* Relais parallel geschaltet, kann der so gelesene und gerade durchgeführte Rechenplan einfach kopiert werden. Relais PP4 wird über Pk1 angesprochen und sobald sich die Werte des Abtasters auf den Pa Relais befinden, wird PP7 geschaltet, wodurch im nächsten Spiel die Werte von Pa auf 'Pl übertragen werden können.

Durch die *Pz* Relais wird der Filmstreifen im Locher bzw. Abtaster um eine Position weiter geschaltet. Wenn durch PP5, PP6 oder PP7 Befehle auf die 'Pl Relais des Lochers geleitet werden, muss ebenfalls *Pz*₁ für die Weiterschaltung des Filmstreifens geschaltet werden. *Pz*₁ schaltet das Folgerelais *Pz*₂, welches auch von PP1 geschaltet werden kann. Ist eines der *PP*₂₋₄ Relais geschaltet oder PP1 mit Kw1 gleichzeitig, wird *Pz*₃ geschaltet. *Pz*₁ schaltet *Pf*₁ und *Pz*₃ schaltet Pf3, welches

die Kupplungsrelais zum Locher und Abtaster darstellen. Diesen sind parallel Relais Pf_2 und Pf_4 mit einem Widerstand geschaltet, um ein durchbrennen der Relais zu verhindern, da diese an Dauerspannung liegen.

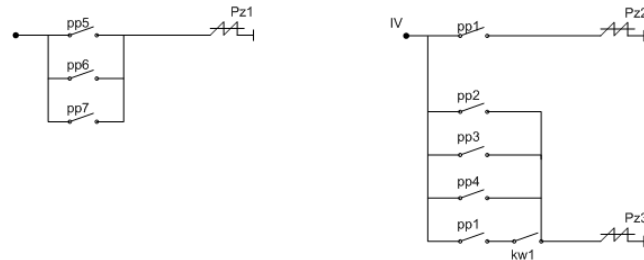


Abbildung 18: Schaltung der Pz Relais

5.4 Eingabe von Dezimalzahlen

Die Eingabe von Zahlen im Dezimalsystem erfolgt über die Eingabe auf dem Tastenfeld. Die dort eingetasteten Ziffern werden durch die Ta Relais gehalten und sobald die Lage des Komma bekannt ist, über die Ta und Tb Relais auf dem Lampenfeld zur Anzeige gebracht. Die Übertragung der Zahl von den Ta Relais auf die Register im Rechenwerk erfolgt über das Steuerrelais Te_1 . Über Te_3 können Ergebnisse und Zwischenwerte aus dem Rechenwerk auf die Ta Relais gegeben werden und von dort visuell durch das Lampenfeld angezeigt werden. Die zusätzlichen Register Tn und Tm können jeweils einen ganzen Wert, wie die Ta Relais aufnehmen. Die Register Tn werden über das Relais Te_1 von den Ta und die Register Tm vom Rechenwerk über Te_4 angesteuert.

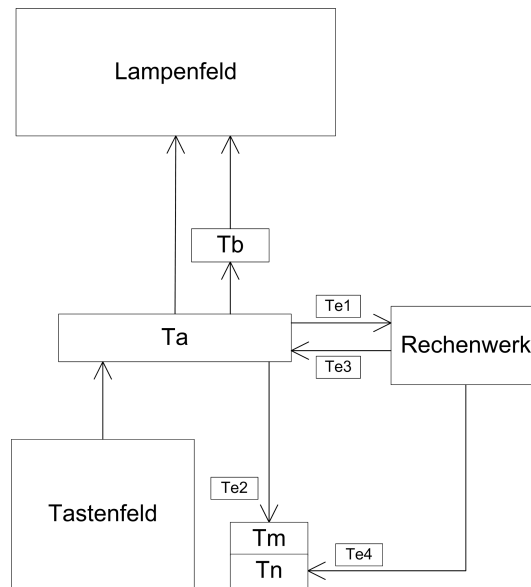


Abbildung 19: Komponenten der Eingabe

Das Eintasten der Zahlenwerte auf dem Tastenfeld führt dazu, dass die Zahlen in die Relaisgruppe Ta übertragen werden. Durch Drücken der Irrtumstaste wird die bis zu dem Zeitpunkt erfolgte Eingabe der Zahl gelöscht und die komplette Zahl muss neu eingetastet werden. Hierfür werden die Ta Relais in die Ausgangsposition versetzt. Die eingetasteten und auf Ta gehaltenen Werte werden dem Benutzer über das Lampenfeld angezeigt. Das Beenden der Eingabe durch die Fertigtaste (Ttf) aktiviert das Te_1 Relais, welches die Zahl an das Rechenwerk leitet. Über Te_3 werden Zahlen vom Rechenwerk auf das Ta Relais gegeben, um z.B. Zwischen- oder Endergebnisse anzuzeigen.

Die logische Umstellung der eingetasteten Zahl vom Dezimal- ins Dualsystem erfolgt über eine automatische Steuerung. Ebenfalls automatisch erfolgt die Umformung vom Dual- ins Dezimalsystem, wenn Werte aus dem Rechenwerk auf die Ta Relais und von dort zur Anzeige gebracht werden. Die genaue Funktionsweise dieser Schaltung wird in den nächsten zwei Kapiteln erklärt.

5.4.1 Eingabe von Zahlenwerten auf die Ta Relais

Die Eingabe der Zahlenwerte erfolgt im Dezimalsystem über die angeschlossene Tastatur oder den Abtaster, der das Einlesen über den Lochstreifen übernimmt. Wobei

über den Lochstreifen in erster Linie Zahlen aus dem Speicherwerk abgefragt werden können oder Konstanten durch Basisoperationen erzeugt werden. Die Eingabe von Zahlenwerten erfolgt in der Regel durch den Bediener über das Tastenfeld. Bei der Eingabe durch die Tastatur gilt nur die Einschränkung, dass jede Zahl maximal fünf von Null verschiedene Ziffern enthalten darf und jede Eingabe mit maximal 10^{-18} oder 10^{+18} als Potenz versehen werden kann. Grundsätzlich besteht auch die Möglichkeit der Eingabe von Sonderwerten oder imaginären Zahlen.



Abbildung 20: Tastatur für Zahlenwerte

Der linke Teil der Abbildung 20 zeigt das Tastenfeld, wie es auch dem Bediener der Z4 vorliegt. Der rechte Teil der Abbildung beinhaltet die technische Bezeichnung der einzelnen Tasten, welche für eine bessere Zuordnung der Tasten zu den verschiedenen Funktionen der Abbildung hinzugefügt ist.

Die Zifferntasten 0-9 sprechen die Relais Tasten Ttz_{0-9} an. Wird kein negatives Vorzeichen durch Ttv kommandiert, wird von einer positiven Eingabe ausgegangen. In der von Zuse gewählten Speicherform wird ein gleitendes Komma verwendet. Es können somit sehr kleine oder auch sehr große Zahlen eingegeben und miteinander verrechnet werden. Durch die Auswahl der geeigneten Potenz über die Tasten $Ttp_{1,2}$ kann diese durch bis zu dreimaliges Drücken den Wert ± 18 annehmen. Der Exponent wird anhand der Entfernung der ersten eingegebenen Ziffern zur Kommastelle berechnet. Folgt der ersten Ziffer sofort das Komma, ist der Wert des Exponenten eins. Jedes Drücken der Taste Ttp erhöht den Exponenten um den Faktor sechs. Die Eingabe des Komma erfolgt an erforderlicher Stelle über Ttk . Bei ganzen Zahlen muss kein Komma getastet werden.

Das Signalisieren eines imaginären Wertes erfolgt durch Tti . Hier ist der Bediener frei, erst die Ziffern und dann Tti oder anders herum zu kommandieren. Die Sonder Tasten Tts_{1-3} signalisieren die Eingabe der Sonderwerte sehr klein, unendlich und

Null.

Vor dem Drücken der Fertigtaste *Ttf* lässt sich die Eingabe mit der Irrtumstaste *Ttr* löschen und die komplette Zahl muss nochmals eingeben werden. Die Übergabe der Zahl von der Tastatur an das Speicherwerk oder das Rechenwerk erfolgt erst, nachdem die Fertigtaste das Ende der Eingabe signalisiert.

Die Tastaturrelais *Ta* besitzen 29 Stellen Ta_{1-29} für die Eingabe, sowie einem zusätzlichem Relais Ta_0 welches eine Belegung der *Ta* Relais anzeigt. Die Eingabe erfolgt vom Bediener im Dezimalsystem und folgt der Konvention $y = 10^a * b$. Innerhalb der Z4 werden die Zahlen allerdings im Dezimalsystem dargestellt und folgen der Konvention $y = 2^a * b$.

Die zur Verfügung stehenden 29 Stellen der *Ta* Relais sind wie folgt aufgeteilt. Das Relais Ta_1 zeigt die Tastung einer Imaginärzahl an. Die Relais Ta_{2-7} sind für den Wert Alpha reserviert und die Relais Ta_{9-28} nehmen den Wert von Beta auf. Ta_8 ist für das Vorzeichen von Beta vorgesehen. Negative Alpha werden durch Zweier-supplement dargestellt. Ta_{29} zeigt an, dass es sich um einen Sonderwert handelt.

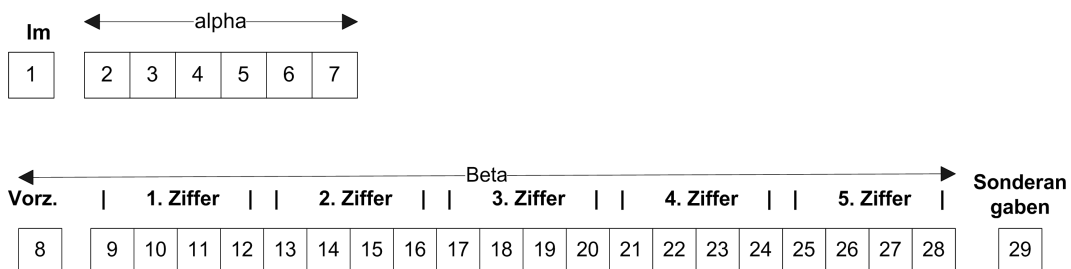


Abbildung 21: Aufteilung der *Ta* Relais

Der Wert Alpha, welcher auf den Relais Ta_{2-7} gehalten wird, unterteilt sich auf die Relais Ta_{2-5} und Ta_{6-7} . Für die verschiedenen Alpha wiederholt sich die Stellung der Ta_{6-7} alle drei Schritte. In den Relais Ta_{2-5} wird der ganzzahlige Teil des Wertes Alpha/3 als Sekundalzahl gehalten. Diese Speicherform ist bei der Umformung des Exponenten aus dem Dezimalsystem ins Dualsystem äußerst hilfreich. Negative Alpha werden als Supplement dargestellt. Mit Ta_{2-7} lassen sich so Alpha von -20 bis $+20$ darstellen.

a	Ta						
	2	3	4	5	6	7	
0	0	0	0	0	0	0	
1	0	0	0	0	0	X	
2	0	0	0	0	X	0	
3	0	0	0	X	0	0	
4	0	0	0	X	0	X	
5	0	0	0	X	X	0	
6	0	0	X	0	0	0	
7	0	0	X	0	0	X	
8	0	0	X	0	X	0	
9	0	0	X	X	0	0	
10	0	0	X	X	0	X	
11	0	0	X	X	X	0	
12	0	X	0	0	0	0	
13	0	X	0	0	0	X	
14	0	X	0	0	X	0	
15	0	X	0	X	0	0	
16	0	X	0	X	0	X	
17	0	X	0	X	X	0	
18	0	X	X	X	0	0	
19	0	X	X	X	0	X	
20	0	X	X	X	X	0	

a	Ta						
	2	3	4	5	6	7	
0	0	0	0	0	0	0	
-1	X	X	X	X	X	0	
-2	X	X	X	X	0	X	
-3	X	X	X	0	X	X	
-4	X	X	X	0	X	0	
-5	X	X	X	0	0	X	
-6	X	X	0	X	X	X	
-7	X	X	0	X	X	0	
-8	X	X	0	X	0	X	
-9	X	X	0	0	X	X	
-10	X	X	0	0	X	0	
-11	X	X	0	0	0	X	
-12	X	0	X	X	X	X	
-13	X	0	X	X	X	0	
-14	X	0	X	X	0	X	
-15	X	0	X	0	X	X	
-16	X	0	X	0	X	0	
-17	X	0	X	0	0	X	
-18	X	0	0	0	X	X	
-19	X	0	0	0	X	0	
-20	X	0	0	0	0	X	

Abbildung 22: Repräsentation von Alpha

Die 20 zur Verfügung stehenden Stellen für Beta werden durch die Relais Ta_{9-28} repräsentiert. Diese sind in fünf Gruppen mit jeweils vier Stellen aufgeteilt. Jede Vierergruppe kann im Dualsystem eine einzelne Ziffer aus dem Dezimalsystem aufnehmen. Dies erklärt auch, warum nur fünf von Null verschiedene Ziffern in der Eingabe verwendet werden können. Für weitere von Null verschiedene Ziffern ist kein Platz vorgesehen.

Ist das Relais Ta_{29} belegt, wird damit das Vorliegen von Sonderwerten angezeigt. Das Sonderwerte eingegeben werden ist unwahrscheinlich, im Rahmen der Rechnung aber als Zwischen- oder Endergebnisse auftauchen und dann angezeigt werden, nicht auszuschließen. Die Relais Ta_{2-28} erhalten dann eine andere Bedeutung, wie es aus Abbildung 23 hervorgeht. S_x auf Stelle 29 zeigt an das Sonderwerte vorliegen, I_x das Vorhanden sein einer Imaginärzahl und über Ta_8 wird das positive Vorzeichen V_x gesetzt.

	Sx	Ix	Vx						
	Ta	29	1	2	3	4	5	6	8
genau Null		X				X	X		
sehr klein		X		X		X			
sehr groß, unbestimmtes Vorzeichen		X			X	X			
sehr groß, imaginär		X	X		X	X			
sehr groß, neg., imaginär		X	X		X				
sehr groß, positiv		X			X				X
sehr groß, positiv, imaginär		X	X		X				X
unbestimmt		X						X	
unbestimmt, imaginär		X	X					X	

Abbildung 23: Sonderangaben

Zu Kontrollzwecken sind die Register Tn und Tm der Konstruktion hinzugefügt. Für die reine Anzeige oder Übertragung der Zahlen sind diese nicht notwendig. Allerdings ermöglichen sie die Überprüfung der richtigen Funktionsweise des Umwandlungsverfahrens vom Dezimal ins Dualsystem und umgekehrt.

Von den Ta Relais kann die Übertragung auf das Rechenwerk über das Te_1 Relais aktiviert werden. Gleichzeitig kann das Relais Te_2 aktiviert werden, welches die Übertragung der jeweiligen Werte von Ta auf Tm ermöglicht. Durch die Aktivierung von Te_2 wird gleichzeitig Tm_0 aktiviert. Tm_0 ist somit nur belegt, wenn sich auf den restlichem Tm Relais ein Wert befindet. Die konkrete Schaltung zur Übertragung von den Ta Relais auf das Rechenwerk oder Tm Register kann in [A.2.1] gefunden werden.

Im Rechenwerk befindet sich nun die eingegebene Zahl im Dualsystem, sowie auf den Tm Relais. Die im Rechenwerk befindliche Zahl wird nun über die Ausgänge des Rechenwerkes $t201-t229$ entweder über Schaltung des Relais Te_3 auf die Ta Relais oder über Te_4 auf die Tn_0 Relais übertragen. Wird Te_3 aktiviert, so wird zusätzlich zur Belegung von Ta_{1-29} auch Ta_0 und mit Te_4 auch Tn_0 aktiviert [A.2.2]. Sind die Werte in den Registern Tm und Tn identisch, funktioniert der Algorithmus zur Umwandlung vom Dezimal- ins Dualsystem und zurück einwandfrei.

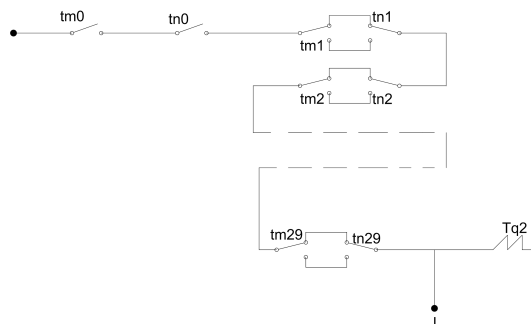


Abbildung 24: Vergleich Tm und Tn Register

Um den Inhalt Tn und Tm zu vergleichen, müssen beide Register mit einer Zahl besetzt sein. Dieses wird durch Tm_0 bzw. Tn_0 signalisiert. Durch diese Relais wird angezeigt, dass sich auf den Relais Tm_{1-29} bzw. Tn_{1-29} ein Wert befindet. Sind also die Relaisgruppen Tn und Tm belegt, können diese miteinander verglichen werden. Die Vergleichsschaltung hat jeweils abwechselnd die Tm und Tn mit gleichen Indizes in Reihe geschaltet und aktivieren das Relais Tq_2 nur, wenn die auf Tm mit Tn

übereinstimmt. Durch das Folgerelais Tq_3 werden die im gemeinsamen Haltekreis befindlichen Relais Tn und Tm wieder abgeschaltet und stehen für eine erneute Übertragung zur Verfügung.

Wird durch die Relais Tm_0 und Tn_0 angezeigt, dass beide Relaisgruppen besetzt sind, das Relais Tq_2 aber nicht anspricht, spricht das Relais Tq_4 an und aktiviert gleichzeitig die Warnlampe L100 und ein Klingelsignal. Beide Werte sollten übereinstimmen, tun dieses aber nicht. Es gibt also einen Fehler bei der Übertragung von Ta auf das Rechenwerk oder vom Rechenwerk auf Ta .

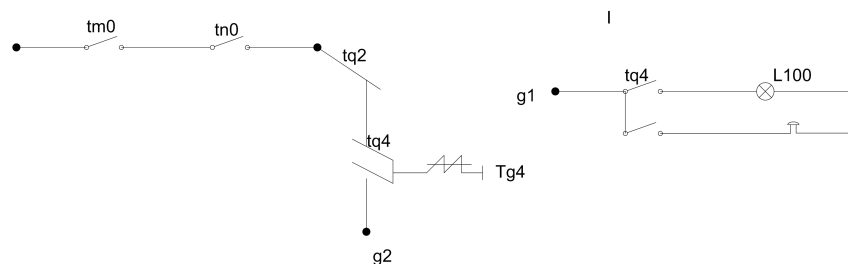


Abbildung 25: Schaltung bei ungleichen Tm und Tn Werten

5.4.2 Anzeige der Eingabe auf dem Lampenfeld

Angezeigt wird die eingetastete Zahl auf dem Anzeigefeld, sobald die Lage der Komma-Stelle bekannt ist. Das Komma wird durch das Tasten von Ttk gesetzt. Bei ganzen Zahlen mit fünf Ziffern ist die Stelle des Komma bekannt, da keine weitere Ziffer mehr getastet werden kann und somit auch kein Komma gesetzt wird. Werden ganze Zahlen mit weniger als fünf Stellen eingetastet, muss die Fertigtaste das Ende der Eingabe signalisieren. Die Zahl wird auf dem Anzeigefeld dargestellt. $Ttp_{1,2}$ und $Tts_{1,2,3}$ bewirken bei Aktivierung das sofortige Anzeigen der Zahl auf der Anzeige. Sobald die Zahl auf dem Lampenfeld angezeigt wird, bewirkt das Drücken der Fertigtaste die sofortige Übertragung auf das Rechenwerk über Te_1 . Das Drücken der Irrtumstaste muss vor Eingabe der Fertigtaste erfolgen, da die eingetastete Zahl dann bereits auf das Rechenwerk übertragen wurde. Die Irrtumstaste löscht die Eingabe auf den Ta Relais und die Zahl muss neu eingegeben werden.

Die Tb Relais sprechen die Lampen 10-79 an [A.1.3]. Diese zeigen die Wertigkeit der eingegebenen Zahl auf dem Lampenfeld an. Entsprechend der Codierung von Alpha auf Ta_{6-7} erfolgt eine Stellenverschiebung der eingegebenen Zahl von den Relais

Ta_{9-28} [A.1.2]. Dies ist notwendig, da das Komma fest auf den Spalten vier, acht und zwölf steht und nicht verschoben werden kann. Die Tb Relais werden über die Stellung der Ta_6 und Ta_7 entsprechend einer Weichenstraße verschoben. Dadurch wird die auf Ta eingetastete Zahl in Abhängigkeit von Alpha auf dem Lampenfeld richtig angezeigt.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
				9	9	9		9	9	9		9		
				8	8	8		8	8	8		8		
<<				7	7	7		7	7	7		7	Potenz überschritten	
∞	-			6	6	6		6	6	6		6		
?	+			5	5	5		5	5	5		5	$10^{(-18)}$	10^{+18}
				4	4	4		4	4	4		4	$10^{(-12)}$	10^{+12}
Im				3	3	3		3	3	3		3	$10^{(-6)}$	10^{+6}
				2	2	2		2	2	2		2		
				1	1	1		1	1	1		1		
			0	0	0	0		0	0	0		0		
				\wedge				\wedge				\wedge		

Löschtaste

Abbildung 26: Lampenfeld

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
				19	29	39		49	59	69		79		
				18	28	38		48	58	68		78		
90				17	27	37		47	57	67		77		99
91	5			16	26	36		46	56	66		76		
93	6			15	25	35		45	55	65		75	86	83
				14	24	34		44	54	64		74	85	82
7				13	23	33		43	53	63		73	84	81
				12	22	32		42	52	62		72		
				11	21	31		41	51	61		71		
		1		10	20	30		40	50	60		70		
		2					3				4			

Trl

Abbildung 27: Lampenfeld

Eine digitale Anzeige, mit Ziffern abhängig von Ihrem Wert, war in der damaligen Zeit noch nicht denkbar. Aus diesem Grund hat Zuse ein Lampenfeld mit 15 Spalten angelegt. Ziffern einer Zahl können nur in den Spalten 3, 5-7, 9-11 und 13 angezeigt werden. Für jede Dezimalstelle der Zahl sind zehn Lampen vertikal verbaut. Beträgt der Wert für eine Ziffer den Wert Null, leuchtet die Lampe auf der passenden Stelle in der untersten Zeile auf. Besitzt die Stelle den Wert neun, leuchtet entsprechend in der obersten Zeile die passende Spalte auf. In Spalte drei, kann nur eine Null angezeigt werden. Es kann pro Spalte immer nur eine Zeile leuchten. Das Komma kann nur in den Spalten vier, acht und zwölf stehen und angezeigt werden. Leuchtet das Komma in Spalte vier, muss automatisch auch die Null in Spalte drei leuchten. Das positive Vorzeichen der angezeigten Zahl wird durch L6, das negative durch

L5 angezeigt. L7 signalisiert das vorliegen einer Imaginärzahl. Die Lampen 10-79 stellen die eingegebene Wertigkeit der Zahl dar. Die Lampen 81-86 zeigen die jeweilige Zehnerpotenz mit 10^6 , 10^{12} , 10^{18} , 10^{-6} , 10^{-12} und 10^{-18} an. Die Lampenfelder 90,91 und 93 zeigen jeweils den Wert „sehr klein“, „sehr groß“ und „unbestimmt“ an. Leuchtet die Warnlampe 99, ist die Potenz überschritten und die Eingabe oder auch Ausgabe ungültig. Der unter dem Lampenfeld befindlichen Knopf Trl lässt das Lampenfeld abschalten.

Exp.												
	20	^										x10^18
	19	^										
	18	^										
	17	^										x10^12
	16	^										
	15	^										
	14	^										x10^12
	13	^										
	12	^										
	11	^										x10^6
	10	^										
	9	^										
+233 000 000	8	2	3	3	^	0	0					x10^6
+23361000	7		2	3	^	3	6	1				
+2336100	6			2	^	3	3	6	1			
+233610	5	2	3	3		6	1	0	^			x10^0
+23361	4		2	3		3	6	1	^			
+2336,1	3			2		3	3	6	^	1		
+281	2		2	8	1	^	0	0				x10^0
+28,106	1			2	8	^	1	0	6			
+2,81	0				2	^	8	1	0	0		
-0,12345	-1	0	^	1	2	3	4	5				x10^0
-0,012345	-2	0	^	0	1	2	3	4	5			
-0,0012345	-3	0	^	0	0	1	2	3	4	5		
-0,00012345	-4		1	2	3	^	4	5				x10^-6
-0,000012345	-5			1	2	^	3	4	5			
-0,0000012345	-6				1	^	2	3	4	5		

Abbildung 28: Darstellung verschiedener Eingaben

Auf Grund der bereits vorgestellten Konvention, dass nur fünf von Null verschiedene Ziffern eingetastet werden dürfen, ergibt sich die Darstellung im o.g. Schema. Es kann im Lampenfeld natürlich immer nur eine Zahl angezeigt werden. Um die Darstellung verschiedener Zahlen darzustellen, werden im o.g. Abbildung verschiedene Zahlen gleichzeitig dargestellt, damit sich ein vollständigeres Bild ergibt.

Nimmt Alpha den Wert Null, sechs, zwölf oder achtzehn an, kann auf der Anzeige eine Ziffer vor dem Komma und bis zu vier Ziffern hinter dem Komma stehen. Besitzt Alpha den Wert eins, sieben, dreizehn oder neunzehn an, können zwei Ziffern vor dem Komma und bis zu drei nach dem Komma stehen. Dieses Muster setzt sich weiter fort, bis für Alpha gleich fünf, fünf verschiedene Ziffern und einer zusätzlichen Null vor dem Komma stehen dürfen.

Es können bis zu fünf von Null unterschiedliche Ziffern vor dem Komma angezeigt werden. Im Beispiel sehen wir für Alpha gleich fünf, dass sechs Ziffern vor einem ver-

meintlichen Komma stehen. Da keine weitere Ziffer mehr eingetastet werden kann, handelt es sich in diesem Fall allerdings um eine ganze Zahl. Die Eintastung erfolgt mit der Eingabe von fünf Ziffern, sowie der Ttp Taste für das Kommandieren einer Potenz mit dem Faktor sechs. In diesen Fällen wird auf der sechsten Stelle eine Null angezeigt, gespeichert werden muss diese nicht. Alpha ist in den gezeigten Fällen ≥ 0 . Für Alpha = 0 befindet sich die erste Ziffer in Spalte sieben und das Komma in Spalte acht. Für negative Alpha wird die Lage des Komma in der Anzeige dahingehend verschoben, dass bis zu drei Stellen vor und zwei Stellen nach dem Komma, sowie dem Faktor der Potenz angezeigt werden. Es kann zu einer Verschiebung kommen, bei der Null Stellen vor dem Komma und fünf Stellen nach dem Komma angezeigt werden. In den Fällen wo Alpha den Wert von -1 bis -3 an nimmt, ist die erste Spalte(3) immer 0 und die Stellen dahinter werden gemäß Anzeige ausgefüllt. Die Anzeige ermöglicht so Zahlen mit Alpha von -18 bis +20 darzustellen. Die Darstellung wiederholt sich alle sechs Alpha, wobei bei größeren und kleineren Alpha die entsprechende Potenz 10^6 , 10^{12} , 10^{18} zusätzlich angezeigt wird.

5.5 Eingabe ins Dualsystem übersetzen

Im Kapitel 5.4 wird die Eingabe der Dezimalzahlen von der Tastatur auf die Ta Relais dargestellt. Von dort erfolgt eine Übertragung der eingegebenen Zahl auf die Register Af, Bf des Rechenwerk durch Kommandierung des Fertig Befehls, welcher das Relais Lu aktiviert und damit die Übersetzung der Eingabe aus dem Dezimalsystem in das Dualsystem auslöst.

Für die Umformung aus dem Dezimalsystem wird jeder einzelne Ziffernwert auf die Relais Bb_{-21} bis Bb_{-24} gegeben und dort ohne Berücksichtigung des Stellenwertes mit zehn multipliziert. Im Dualsystem wird dies erreicht, durch die Bildung der Summe der Verschiebung um eine und um drei Stellen. Die Verschiebung um eine Stelle entspricht der Multiplikation mit zwei, die um drei Stellen der mit acht. Werden beide Werte addiert, ist der Zahlenwert mit zehn multipliziert. Nach der Multiplikation prüft eine Schaltung, ob noch weitere von Null verschiedene Ziffern vorhanden sind. Ist dieses der Fall wird diese auf Bb_{-21} bis Bb_{-24} übertragen und die gesamte Zahl wieder mit zehn multipliziert.

Diese Art der Umformung ist nur für ganze Zahlen anwendbar, weshalb Zuse für jede Umformung die Eingabe ohne Komma betrachtet und die Zahl entsprechend dem Verfahren umformt. Alpha wird im ersten Spiel auf das Additionswerk A übertragen und muss am Ende um die Anzahl der übertragenen Ziffern verkleinert werden, da für jede übertragene Ziffer eine Multiplikation mit zehn durchgeführt wurde. Daher wird während der Multiplikation mit zehn, der Wert Alpha gleichzeitig um eins erniedrigt. Am Ende muss die Mantisse noch mit zehn multipliziert werden, bis der Exponent ein Vielfaches von drei ist. Dann wird die Mantisse ausgerichtet, dass sie der Konvention $1,0 \leq b < 2,0$ entspricht und die Übersetzung ist abgeschlossen.

Das Relais Te_1 , welches die Übertragung des auf der Tastatur befindlichen Wertes veranlasst, wird über die Bedingung $Tf_4 \wedge Tf_5$ aktiviert. Tf_4 zeigt, dass die Tastatur Relais belegt sind. Tf_5 zeigt den Status des Rechenwerks an. Wenn es geschaltet ist, ist das Eingaberegister im Rechenwerk frei und kann eine Zahl von der Tastatur aufnehmen.

Es erfolgt nun die Übertragung des auf Ta befindlichen Wertes in die Register Af, Bf des Rechenwerk. Dazu werden verschiedene Zwischenschritte notwendig, welche durch den Schrittschalter Ud_1 nacheinander angesteuert werden. Der Schrittschalter Ud_1 verlangt die Aktivierung der Relais lu und ln_2 . Lu ist der Übersetzenbefehl und durch ln_2 wird signalisiert, dass gerade keine Rechenoperation im Rechenwerk abläuft. Im ersten Spiel wird das Relais Ea_1 aktiviert, welches die Übertragung des Exponenten vom Register Af auf Aa steuert. Nun wird durch den Schrittschalter in jedem Spiel abwechselnd das Relais Ub gemeinsam mit Fz_1 bis Fz_5 für die jeweilige Ziffer aus dem Dezimalsystem aktiviert und im nächsten Schritt Ua .

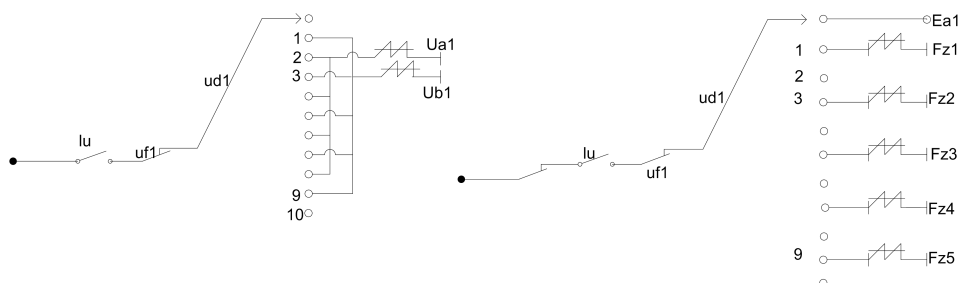


Abbildung 29: Übertragung der Eingabe auf das Rechenwerk

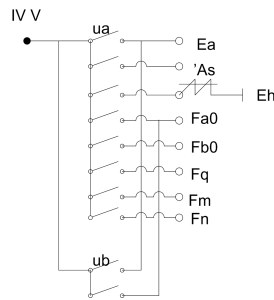


Abbildung 30: Übertragung der Eingabe auf das Rechenwerk

Bei jeder Übertragung und Aktivierung von Ub werden die Relais Ea_0 und Fa_0 aktiviert, welche jeweils einen blinden Umlauf über das Additionswerk veranlassen. Bei jeder Multiplikation werden von Ua folgende Relais aktiviert: Ea_0 , $'As$, Eh , Fa_0 , Fb_0 , Fq , Fm und Fn . Ea_0 wiederholt den blinden Umlauf des Wertes Alpha über das Additionswerk. Es wird sichergestellt, dass der Wert Alpha damit gehalten wird. $'As$ aktiviert das Additionswerk welches die Register Ba und Bb addiert und damit die Multiplikation von zehn durchgeführt hat. Das Ergebnis wird auf Be gegeben. Eh subtrahiert eins vom Wert Alpha. Da dieser dreistellig ist, also die Belegung von LL auf den letzten beiden Stellen nicht vorkommt, sondern von $0L0$ auf $L00$ gesprungen wird, wenn dieser um eins erhöht wird und umgekehrt. In Abhängigkeit von ae_1 und ae_0 wird bei der Schaltung von Eh und einem Wert auf ae_1 oder ae_0 L abgezogen, stehen auf ae_1 und ae_0 keine Werte müssen $L0$ (Schaltung Ab_1) von Alpha abgezogen werden. Abbildung 31 Fa_0 erzeugt einen blinden Umlauf im Additionswerk B mit Ziel der Ba Relais. Fb_0 erzeugt einen blinden Umlauf im Additionswerk B mit Ziel der Bb Relais. Fq lässt den Wert über die kleine Weichenstraße und bei Fq „ja“ und Fp „nein“ um eine positive Stelle verschieben. Fm und Fn steuern die große Weichenstraße und veranlassen eine Verschiebung um drei Stellen auf die Bb Relais.

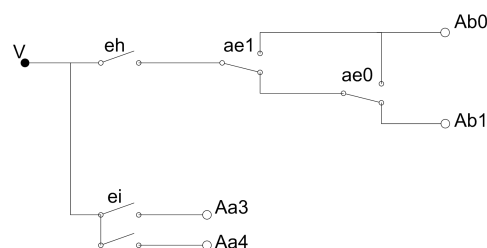


Abbildung 31: Bildung von Alpha

Durch den Durchlauf über die Weichenstraße erhalten wir somit einmal den eingegebenen Wert um eine Stelle verschoben, also mit zwei multipliziert. Durch die

Verschiebung von drei Stellen erhalten wir den Wert der Eingabe mit acht multipliziert. Eine Addition der beiden Werte hat somit die Multiplikation von zehn im Dualsystem erreicht. Mit der Multiplikation mit zehn, muss nun Alpha um eins verringert werden, da der um zehnfach höhere Wert in der Mantissee gehalten wird. Alpha wird zu diesem Zeitpunkt noch im Dezimalsystem geführt. Somit wird der Wert der Mantissee jedes Mal mit zehn multipliziert und gleichzeitig über *Eh* Alpha um eins verringert.

Der Schrittschalter *Ud₁* geht so durch alle zu übertragenden Ziffern. Gleichzeitig wird überprüft ob noch eine von Null verschiedene Ziffer übersetzt werden muss. Sind alle Zahlen übertragen oder die noch zu übertragenden Ziffern besitzen als Wertigkeit eine Null, dann wird das Relais *Uf₁* geschaltet. In Abhängigkeit der Werte auf den Relais *ae₀* und *ae₁*, auf welchem sich der Alpha Wert anteilig befindet, wird das Relais *Ua* für eine weitere Multiplikation mit zehn aktiviert oder das Relais *Uc₁*. Eine weitere Multiplikation mit zehn ist notwendig, wenn die Werte auf *ae₀* und *ae₁* nicht Null sind und sich es sich damit bei Alpha um kein vielfaches von drei handelt. Ist Alpha ein Vielfaches von drei, spricht *Uc₁* an. Alpha ist somit durch drei teilbar. *Uc₁* ist Teil der Relaiskette *Uc₁*, *Uc₂*, *Uc₃* und bewirkt die Ausrichtung der Mantissee, da diese kleiner als 1,0 sein kann. *Uc₁* aktiviert die Relais *Ff₁* und *Ea₀*. Über *Ff₁* wird der Wert der letzten Addition somit auf das Register *Bf* übertragen. Über *Ea₀* erfolgt ein blinder Umlauf, welcher den Wert von Alpha auf *Aa* überträgt.

Uc₃ als letzter Schritt der Relaiskette aktiviert *Fb₁*, *Fe₂*, *Ei* und *'As*. *Fb₁* aktiviert die Übertragung der Mantissee auf *Bb* über die große Weichenstraße. *Fe₂* zeigt die positive Verschiebung über die große Weichenstraße an. *Ei* aktiviert die Register *Aa₃* und *Aa₄* mit belegt diese jeweils mit einer eins. Um zu verhindern, dass nach oder während des Ausrichtens eine Übertragung eines Wertes oder dessen Multiplikation durchgeführt wird, das Relais *Uf₂* geschaltet. Dadurch ist ein erneutes aktivieren von *Ua₁* oder *Uc₁* unmöglich.

Alpha	Ae							Um	Ud		Ab										Alpha
	5	4	3	2	1	0					7	6	5	4	3	2	1	0			
18		x	x					6	3			x	x	x		x	x		LLL0LL		
15		x		x				5	3			x	x				x		LL000L		
12		x						4	3			x				x	x	x	L00LLL		
9			x	x				3	2				x	x	x		x		LLL0L		
6			x					2	2				x			x	x		L00LL		
3				x				1	2					x			x		L00L		
-3	x	x	x	x				-1	4			x	x	x		x	x		-L0L0		
-6	x	x	x					-2	4			x	x		x	x			-L0L000		
-9	x	x		x				-3	4			x	x				x		-LLLL0		
-12	x	x						-4	5			x		x	x				-L0L000		
-15	x		x	x				-5	5			x			x	x	x		-LL00L0		
-18	x		x					-6	5			x				x			-LLLL00		

Abbildung 32: Umwandlung von Alpha

Die letzte Korrekturmultiplikationen von Alpha mit 10^n wird durch U_{c3} veranlasst. Durch U_{c1} wird der Alpha Wert, welcher nun ein ganzes durch drei teilbar ist, auf Ae übertragen. In Abhängigkeit der Belegung von Ae_2 - Ae_5 ist der Alpha Wert codiert. Wird die Annahme getroffen, dass Ae_2 2^0 bis Ae_4 2^2 entspricht, kann der Wert einfach abgelesen und mit drei multipliziert werden, um Alpha zu erhalten. Jedem Alpha ist jeweils ein Um Relais zugeordnet, jedem Um Relais zusätzlich ein passender Schrittschalter Ud_{2-5} , welcher die eigentliche Multiplikation durchführt. Die Um Relais steuern die Belegung der Ab Relais. D.h. durch die Aktivierung eines Um Relais kann der dazugehörige Alpha Wert im Dualsystem auf den Ab Relais eingestellt werden. In der übersetzten Darstellung entspricht 10^3 aus dem Dezimalsystem der Codierung von $2^{1001} * 1,1111010000000000000000$ im Dualsystem. Für den Exponenten sind alle Potenzen von -18 bis +18 in dreier Schritten mechanisch codiert und müssen nicht einzeln umgerechnet werden. Die Umwandlung des Exponenten kann in einem Takt erfolgen.

Über U_{c3} wird das Relais Un_0 aktiviert und damit die Relaiskette Un_1 , Un_2 , Un_3 aktiviert. Un_1 aktiviert die Relais Ea_1 , Fa_1 und Fb_1 . Un_1 aktiviert die Relais Ea_0 , Fa_0 und Fb_0 . Ist das Relais ae_5 belegt, handelt es sich um einen negativen Exponenten, dann wird zusätzlich das Relais Fq aktiviert, welches eine Verschiebung auf der kleinen Weichenstraße von +1 steuert. Hierdurch erfolgt die Übertragung von der Werte von Ae, Be auf die Aa, Ba Register.

Un_3 aktiviert mit Ea_0 und Fb_0 den blinden Umlauf in beiden Additionswerken von Wert a und b. Ist nun Un_1 aktiviert, wird der durch die Belegung der Ae_{2-5} gehaltene Exponent und die Steuerung des passenden Um Relais auf die Register Ab

übertragen. Damit befindet sich der Wert von Alpha für die Korrekturmultiplikation ebenfalls in den Registern Ab. Der so passende Schrittschalter Ud_{2-5} führt nun die eigentliche Korrekturmultiplikation durch.

Ist Un_1 aktiviert, wird in Register Ab in Abhängigkeit der $Um_{-6..+6}$ Relais der Wert Alpha dargestellt. Die Zuordnung der Schrittschalter $Ud_2 - 5$ erfolgt entsprechend der Abbildung. Die Korrekturmultiplikation wird durch die Schrittschalter Ud_{2-5} gesteuert. Diese Steuern gleichzeitig die Weichenstraße und das Additionswerk.

5.6 Lampenfeld

Das „Cursor“ gesteuerte Lampenfeld wurde erstmalig für die Z4 entwickelt und eingesetzt. Bei längeren Rechenplänen und einer Eingabe von verschiedenen Variablen, soll der Benutzer bei der Eingabe der verschiedenen Variablen in die Z4 unterstützt werden. Gerade bei langwierigen Rechnungen mit einer großen Anzahl an Variablen, soll eine Verwechslung bei der Eingabe ausgeschlossen werden.

Das Lampenfeld besteht aus 50 Feldern, von denen immer nur eins aufleuchten kann. Es befindet sich ein imaginärer Cursor unter dem Lampenfeld, welcher durch das Lampenfeld nach rechts, links, oben und unten bewegt werden kann. Das Feld, auf welchem sich der Cursor befindet, leuchtet auf. Es können alle 50 Felder nacheinander angesteuert werden. Zu Beginn befindet sich der „Cursor“ über dem Feld 301, also auf dem nicht vorhandenen Feld 300. Über dem Lampenfeld ist eine durchsichtige Mattscheibe angebracht, welche es dem Bediener ermöglicht, einen Rechenplan oder ein Blatt Papier mit entsprechender Beschriftung auf dem Lampenfeld zu platzieren.

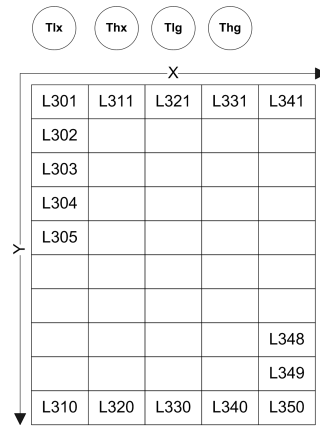


Abbildung 33: Lampenfeldanzeige

Die Abbildung zeigt den schematischen Aufbau des Lampenfeldes im Schaltpult. Es sind 50 Lampen in fünf Spalten und zehn Zeilen angeordnet. Zu Beginn befindet sich der „virtuelle“ Zeiger über der Lampe 301, es leuchtet somit keine Lampe auf. Mit den Befehlen über die Taste Tlx bewegt sich dieser Zeiger in X Richtung, durch drücken der Taste Tly in Y Richtung. Die Weiterschaltung der auszuwählenden Lampe ist mit jeweils einem Schrittschalter Pd_1 für die X und Pd_2 für die Y Richtung realisiert. Über Thx und Thy lassen sich die Schrittschalter über einen Heimlaufkreis wieder in die Ausgangsposition bringen. Alle vier Befehle können natürlich auch über den Lochstreifen kommandiert werden. Wird der Zeiger nun mit dem Befehl Tly von seiner Ausgangsposition auf die Lampe 301 bewegt, leuchtet diese.

Voraussetzung für die Steuerung der Lampen, ist die Aktivierung des Ph_2 Relais, welches erst nach Aktivierung von Ph_1 und $\overline{Pb_0}$ geschaltet wird. Ist dieses Aktiviert, werden über die Schrittschalter Pd_1 und Pd_2 und dadurch die passende Lampe mit Strom versorgt, sie leuchtet. Ph_2 Es bleibt bis zum Ende des Rechenplans aktiviert. Um eine Veränderung des Cursors in Y Richtung zu erreichen wird der Schrittschalter Pd_1 um einen Schritt weiter geschaltet. In X Richtung der Schrittschalter Pd_2 .

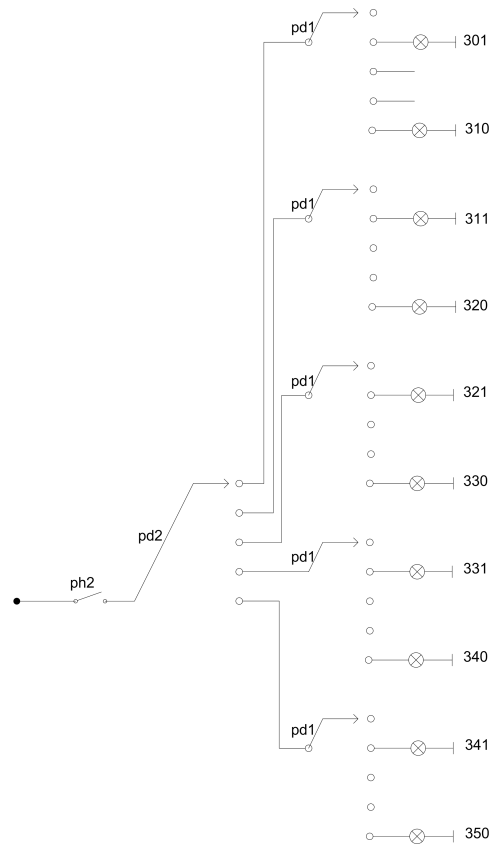


Abbildung 34: Steuerung des Lampenfeldes durch Schrittschalter

Die Schrittschalter sind über die Heimlaufkreise Pu_1 für Pd_1 und Pu_2 für Pd_2 verbunden. Ist allerdings Ck_2 aktiviert und signalisiert damit einen ablaufenden Rechenplan, werden die Schrittschalter über Ph_1 und Pb_1 gesteuert. Ph_1 steht wiederum in Abhängigkeit der $Pb_{2,4,5,10-12}$ Relais.

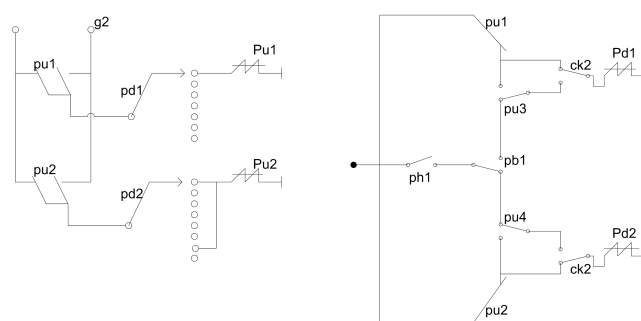


Abbildung 35: Steuerung der Schrittschalter

Für den Einsatz des Lampenfeldes sind mehr Szenarien denkbar:

Sind für eine Rechnung die Eingaben von mehreren Variablen notwendig, kommt es hierbei auf die richtige Reihenfolge der eingegebenen Variablen an. Eine Verwechslung sollte vermieden werden. Durch die Nutzung des Lampenfeldes soll es dem

Bediener erleichtert werden, die richtige Variable zu finden. Es muss nun nicht mehr von Hand ein Rechenplan mit Notizen versehen werden, welche Variable bereits eingegeben wurde und welche als nächstes einzugeben ist. Durch die Auflage eines Plans mit der Bezeichnung der einzugebenden Variablen, können diese nacheinander angeleuchtet werden. Leuchtet nun ein Feld, welches nach dem Rechenplan mit einem X versehen ist, weiß der Bediener dass die Eingabe dieser Variablen erforderlich ist. Ein abstreichen, dass das X bereits eingegeben wurde und nun das Y einzugeben ist entfällt. Allerdings ist einmalig ein höherer Programmieraufwand des Rechenplans mit Eingabe der Lampenfeldbefehle notwendig. Eine Verwechslung der Variablen bei der Eingabe wird hiermit verringert.

Besteht eine Formel aus den Variablen x, y, z kann auf das Papier, welches auf dem Lampenfeld aufgelegt ist, das Feld 301 mit X, Feld 302 mit Y und Feld 303 mit Z bezeichnet werden. Während des Ablaufs der Rechnung ist nun zuerst die Eingabe der Variablen X notwendig. Dafür wird in den Programmablauf der Cursor von der Warteposition auf das Feld 301 bewegt. Der Bediener weiß nun, dass beim Aufleuchten dieses Feldes, der Wert für die Variable X einzugeben ist. Im Laufe der Rechnung wird das Feld 302 beleuchtet und der Bediener kann von dem erleuchteten Papier ablesen, dass eine Eingabe des Wertes Y notwendig ist. Bei einfachen Rechnungen ist dieses Verfahren aufwendig und nicht unbedingt notwendig, mit steigender Variablenzahl hilft es aber dem Bediener den Überblick zu behalten.

Ebenfalls kann das Lampenfeld für eine Zuordnung der Ausgaben verwendet werden. Erfordert eine Rechnung die Ausgabe von mehreren Zwischen- oder Endergebnissen, kann mit Hilfe des Lampenfeld die Zuordnung der Ausgabe verbessert werden. Ist das Ergebnis einer Rechnung der Wert x oder der Zwischenwert k, kann ein entsprechendes Feld beleuchtet werden und der Bediener kann die Ausgabe gleich der richtigen Variablen bzw. Gleichung zuordnen. Verwechslungen werden auch hier reduziert.

5.7 Architektur des Speichers

Das Vorgängermodell, die Z3 verwendete einen Speicher vollkommen aus Relais konstruiert. Diese benötigen viel Platz und sind kostenintensiv. Die gemachten Erfahrungen mit dem mechanischen Speicher der Z1 veranlassten Zuse, auf diesen zurück

zu kommen. Dieser mechanische Speicher benötigt eine deutlich kleinere Grundfläche und ist kostengünstiger.

Für die Z4 wurden in der ersten Ausbaustufe 32 Bit lange Zahlenwerte gespeichert und 64 Speicherzellen realisiert. Der größte Vorteil liegt in der relativ einfachen Konstruktion, welche durch ähnliche Bauteile, vor allem selbstgestanzten Blechen, aufgebaut ist. Dieses war insbesondere für eine Serienfertigung, welche sich Konrad Zuse vorstellte und den zu erwartenden Kosten der Produktion wegweisend. Für die Prototypen wurden die Speicher allerdings noch von Hand produziert. Der mechanische Speicher der Z4 wurde nach dem Krieg noch acht Jahre genutzt, was dessen Zuverlässigkeit unterstreicht.

Der Speicher der Z4 ist mit dem der Z1, mit Ausnahme der Bit Länge sehr ähnlich. Daher findet sich eine detailliert Beschreibung des mechanischen Speichers in der von Konrad Zuse verfassten Patentschrift Nr. 924107[8], sowie den Arbeitspapieren der GMD 321[9].

Im Folgenden soll insbesondere auf die Steuerung des Speichers durch das Planwerk eingegangen werden. Durch die universelle Konstruktion kann jede andere mechanische Speicherkonstruktion verwendet werden.

Auf Grund mangelnden Materials und Geld ist bis 1945 nur ein Speicher mit 64 Speicherzellen realisiert worden. Durch den modulare Aufbau ist es aber zu jedem Zeitpunkt möglich, diesen auf die bis zu 1024 geplanten Speicherzellen zu erweitern. Der Speicher von 64 Speicherzellen repräsentiert eine von acht mögliche Zellengruppen. Welche zusammengefasst jeweils eins von zwei Speicherwerken darstellen. Da Speicherwerke und Zellengruppen bei jeder Speicheroperation explizit ausgewählt werden müssen, wird bei 64 Speicherzellen einfach nur mit einer Zellengruppe gearbeitet.

Die einzelnen Speicherzellen können direkt von der Tastatur und vom Rechenwerk angesprochen und mit Inhalten gefüllt werden. Wobei das Planwerk die Aus- und Anwahl der zu belegenden oder auszulesenden Speicherzelle übernimmt. Da es sich um einen mechanischen Speicher handelt, wird eine einmal gespeicherte Zahl so lange im Speicherwerk gehalten, bis eine neue Zahl auf dieselbe Speicherzelle

gegeben wird. Der Inhalt wird mit dem neuen Belegen eines Wertes automatisch überschrieben. Dies hat zur Folge, dass bei manchen Rechnungen oder beim Start eines neuen Rechenplanes erstmal der gesamte Speicher gelöscht werden muss, um nicht fälschlicherweise mit alten Speicherwerten zu arbeiten. Eine einmal gespeicherte Zahl kann beliebig oft an das Rechenwerk übertragen werden.



Abbildung 36: Cache

In Register C_n wird vom Planwerk die Speicheradresse der anzusprechenden Speicherzelle gegeben. Ist das Register C_n noch belegt, was durch C_{n20} angezeigt wird, wird die Adresse der Speicherzelle erst auf C_b und sobald C_n wieder frei ist, von C_b auf C_n gegeben. Das C_n oder C_b bereits mit einer Adresse belegt sind, wird durch C_{n20} bzw. C_{b20} angezeigt. Auf den 32 Bit langen C_a und C_c Relais wird die Zahl gehalten. Erfolgt ein Speicherbefehl überträgt das Rechenwerk die Zahl auf die C_a Relais, durch C_{a0} wird angezeigt, dass diese frei sind. Von den C_a Relais erfolgt die mechanische Einstellung im Speicher. Erfolgt ein Ablesebefehl durch Pb_{10} kommandiert, wird die Zahl vom Speicher auf die C_c Relais gegeben und von dort auf das Rechenwerk übertragen. Auch hier zeigt C_{c0} an, dass die Register frei oder belegt sind.

Eine Kontrolle auf einwandfreie Funktion des Speicherwerks kann ohne direkten Anschluss an die Z4 erfolgen. Die einzelne Speicherzelle wird über die Schalter Tcn_{0-8} angewählt und damit die Steuerleitungen C_{10-18} angesprochen. Die Schaltung bewirkt, dass die Register $C'_{n'_{0-8}}$ belegt werden und die Adresse für die anzuwählende Speicherzelle halten. Über die Schalter Tca_{1-32} wird das Register Ca'_{1-32} über die Steuerleitungen $C_{101-132}$ mit der zu speichernden Zahl belegt. Über die Lampen $L_{201-232}$ kann der Inhalt der C_c Relais angezeigt und die richtige Funktionsweise des Speichers überprüft werden. Die C_c Relais nehmen bei einer Ableseoperation den Wert aus dem Speicherwerk auf und halten diesen für das Rechenwerk in Bereitschaft. Die Drucktasten Tcs und Tcr simulieren jeweils die Kommandierung eines

Speicher- bzw. Ablesevorgangs im Speicherwerk.

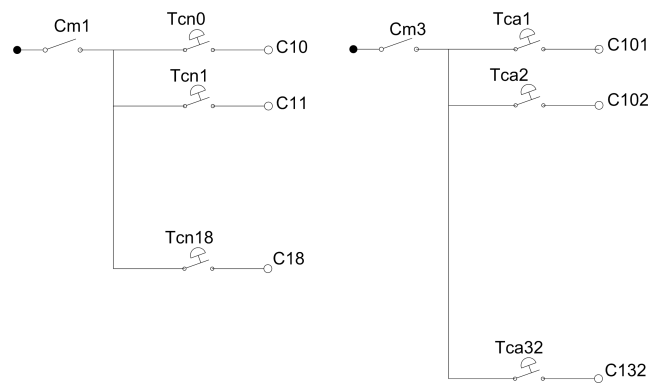


Abbildung 37: Manuelle Schaltung des Speichers

Die Steuerung des Speicherwerks durch das Planwerk gestaltet sich wie folgt. Die erste Unterscheidung, bei Speicherbefehlen, erfolgt in zwölf- oder achtstellige Befehle für die Anwahl der Speicherzelle. Ist Pm_3 nicht aktiviert und zeigt damit das Vorliegen von achtstelligen Befehlen an, muss nur Pb_{12} belegt sein, was eine achtstellige Belegung der Pb Relais des Planwerkes signalisiert. Ist Pm_3 aktiviert, muss zusätzlich noch Pb_{13} signalisieren, dass die komplette zwölfstellige Speicheradresse auf den Pb Relais liegt. In beiden Fällen muss Pb_{11} das Anzeigen eines Speicherbefehls signalisieren. Pb_{10} unterscheidet in Ablesebefehl Cr oder Speicherbefehl Cs .

Bei einem Ablesebefehl welcher Cr aktiviert, muss sicher gestellt sein, dass sich auf den Cc Relais, welche den Wert der Speicherzelle auf das Rechenwerk weiter leiten, noch kein Wert steht. Dieses wird durch ein nicht aktives Cc_0 Relais angezeigt und Cr aktiviert. Liegt bereits ein Wert auf den Cc Relais, kann das Rechenwerk durch die Aktivierung von lz_1 anzeigen, dass die auf den Cc Relais befindlichen Werte auf das Rechenwerk übertragen werden und bis zur vollzogenen Ablesung des neuen Wertes aus dem Speicher, die Cc Relais wieder frei sind.

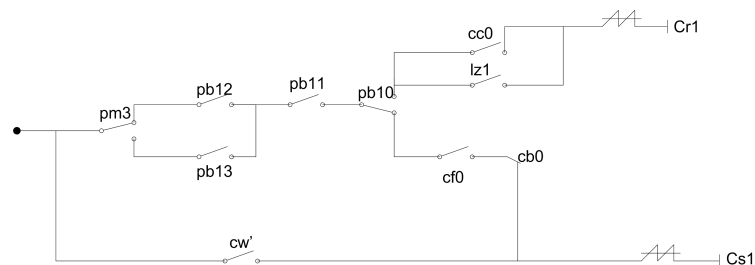


Abbildung 38: Schaltung eines Speicherbefehls

Bei einem Speicherbefehl welcher Cs aktiviert, muss das Rechenwerk durch Cf_0 signalisieren, dass die Rechnung bereits beendet ist und ein Wert für die Übertragung auf die Ca Relais bereit steht. Alternativ kann der Speicherbefehl Cs durch Cw' geschaltet werden. Wären die Cn Relais belegt, wird die Adresse der Speicherzelle auf Cb geleitet. Ist dies der Fall, wird die Speicherzelle nicht mehr direkt durch die Pb Relais angewählt, sondern durch den auf den Zwischenrelais Cb befindlichen Wert angesteuert.

Das Folgerelais Cr_2 welches nachfolgend zu Cr_1 schaltet, aktiviert Cc_0 und zeigt damit das Vorhandensein eines Wertes auf den Cc Relais an. Dadurch wird verhindert, dass zwei Ableseoperationen in Folge die Cc Relais überschreiben, bevor der mit Cr angeforderte Wert auf das Rechenwerk übertragen worden ist. Wird der Wert von Cc auf das Exponentenregister Af oder Ag im Rechenwerk übertragen, durch die Aktivierung der Relais Ef_2 oder Eg_2 angezeigt, wird gleichzeitig Cc_0 deaktiviert. Damit ist der Wert vom Speicher auf das Rechenwerk übertragen und es kann ein neuer Wert vom Speicherwerk abgelesen und auf die Cc Relais gelegt werden.

Sind Cr_1 , Cs_1 , Cb_{19} oder $'Pf_5$ geschaltet, wird dadurch Pf_5 aktiviert, welches die Pb Relais deaktiviert. Der Speicher- oder Ablesebefehl wurde mit der Speicheradresse an den Speicher übermittelt und die Pb Relais stehen für die Aufnahme eines neuen Befehls bereit, während das Speicherwerk noch Ablese- oder Speicheroperation ausführt. $'Pf_5$ wird durch einen der Befehle Start, Schluss, Umstellen oder Lampenfeld geschaltet.

Die Übertragung der Pb Befehle auf das Speicherwerk, kann entweder direkt von Pb auf Cn oder über den Umweg der Cb Befehle erfolgen. Die Übertragung von Pb direkt auf die Cn Relais erfolgt, wenn durch Pb_{10} angezeigt wird, dass es sich um einen Ablesebefehl handelt und durch Cc_0 angezeigt wird, dass sich auf Cc noch kein Wert befindet. Handelt es sich um einen Speicherbefehl, durch Pb_{10} angezeigt, muss durch Cf_0 angezeigt werden, dass das Rechenwerk die Rechnung beendet hat und der Wert vom Rechenwerk auf die in Pb gehaltene Speicheradresse übertragen werden kann. Des Weiteren muss Cb_{20} signalisieren, dass die Cb Relais nicht belegt sind und der Wert des Rechenwerks auf die Adresse der Pb Relais übertragen werden darf.

Liegt ein Speicherbefehl vor, das Rechenwerk hat allerdings noch keinen Wert berechnet und die Cb Relais sind nicht belegt $\overline{Cf_{20}}$, wird die Adresse der Speicherzelle von Pb auf Cb gegeben.

Liegt ein Speicherbefehl vor, auf den Cb Relais befindet sich die Adresse der Speicherzelle und das Rechenwerk hat seinen Wert auf die Ca Relais gegeben oder die Rechnung soeben beendet und ist bereit den Wert auf das Speicherwerk zu geben Cf_0 , wird dieses über die Aktivierung von Cw angezeigt. Cb überträgt nun seinen Inhalt auf die Cn Relais, welche das Wählwerk des Speichers mit der Speicheradresse einstellt.

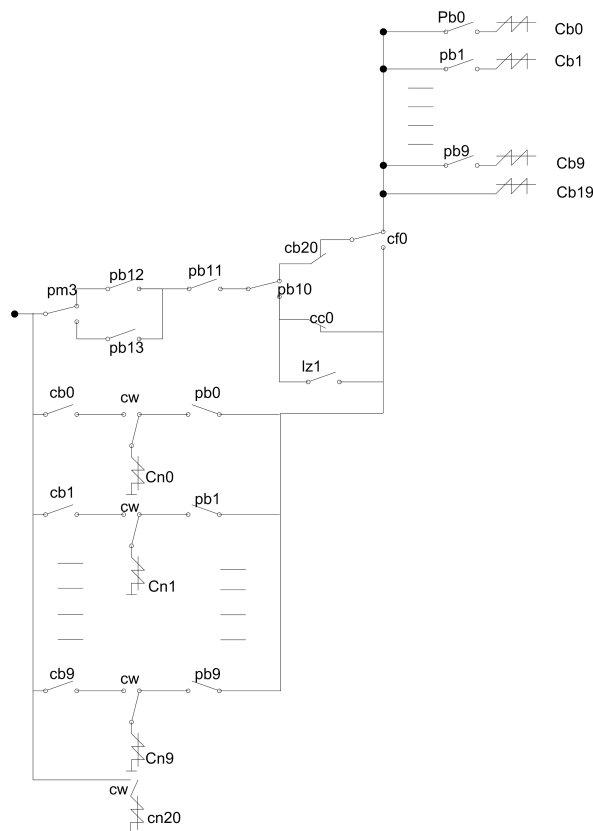


Abbildung 39: Schaltung zur Übertragung der Speicheradresse

Der Umweg über die Cb Relais ist vorgesehen, wenn durch Cf_0 angezeigt wird, dass das Rechenwerk die Rechnung noch nicht beendet hat, aber bereits eine Adressierung des Speichers durch die Pb Relais erfolgt ist und gleichzeitig noch eine Speicheradresse auf den Cn Relais liegt. Die Cb Relais dienen somit als Puffer der Cn Relais für die Adressierung der Speicherzelle.

Die Werte von Pb_{0-9} werden entweder auf die Cb_{0-9} Relais gegeben und aktivieren damit auch Cb_{19} oder auf die Cn_{0-9} Relais. Erfolgt die Übertragung auf die Cn Relais über Cb wird zusätzlich noch Cn_{20} aktiviert.

Wird Ef_2 oder Eg_2 geschaltet, wird damit der Ausgang C1 geschaltet, welcher entsprechend der Belegung der CC Relais die Ausgänge vom Speicherwerk C201 bis C232 schaltet und damit den Wert der Speicherzelle auf den Eingang des Rechenwerks legt.

6 Arithmetische Algorithmen

6.1 Addition

Die Operationen einer Addition sind recht simpel und verlangen keinen Schrittschalter, wie er bei komplizierteren Schaltungen zum Einsatz kommt. Die Addition wird in Form einer Relaiskette, durch die abwechselnde Aktivierung der *Sa* und *Sb* Relais durchgeführt. Die *Sa* Relais führen die jeweiligen Operationen durch und die *Sb* sorgen für die Weiterschaltung innerhalb der Relaiskette.

Die einfache Addition ist in drei Spielen durchführbar. Im ersten Spiel wird die Differenz der Exponenten der beiden Summanden gebildet. Die Mantisse des kleineren Summanden wird über die große Weichenstraße auf das Register Bb geleitet. Die Weichenstraße steuert entsprechend der ermittelten Exponentendifferenz die Verschiebung der Mantisse. Die Register Ba und Bb besitzen 26 Stellen hinter dem Komma. Eine Verschiebung um mehr als 24 Stellen könnte die Mantisse Null werden lassen, da die Operandenregister Bf und Bg nur 24 Stellen hinter dem Komma speichern können. Daher können nur Zahlen summiert werden, welche in ihrer Größenordnung eine Exponentendifferenz von weniger als 24 haben. Durch die Verschiebung der Mantisse müsste der Exponent der kleineren Zahl um den entsprechenden Wert erhöht werden. Einfacher ist es, gleich den größeren Exponenten der beiden Summanden zu verwenden.

Die Mantisse des größeren Summanden wird ohne Stellenverschiebung auf Ba gegeben. Es befindet sich nun jeweils eine Mantissen in Ba und Bb. Die Addition beider Register ist Dank der von Zuse entwickelten Carry-Look-Ahead Schaltung in einem Takt möglich. Der Exponent beider Summanden ist durch die vorhergehende Stellenverschiebung gleich und muss daher nicht verändert werden. Im zweiten Takt muss nur das Additionswerk B die beiden Mantissen addieren. Beide Summanden können durch das Vorzeichen angezeigt, positiv oder negativ sein. Ebenfalls gibt es vier Additions- und Subtraktionsbefehle: „-a-b“, „b-a“, „a-b“, „a+b“. In Abhängigkeit des gewählten Befehls und der Summanden erfolgt nun die Addition bzw. Subtraktion. Handelt es sich um eine Addition, kann die Mantisse einen Wert größer zwei annehmen und muss im dritten Takt nur noch ausgerichtet und der Exponent entsprechend der Verschiebung erhöht werden. Die Addition ist beendet.

Handelt es sich um eine Subtraktion und das Ergebnis ist negativ muss noch das

Komplement erzeugt werden. Ist die Mantisse bei der Subtraktion kleiner als 1,0 geworden, muss diese zusätzlich ausgerichtet werden.

Die Relaiskette wird durch den Anschluss l2 angestoßen, wenn durch die durch das Operationsrelais Ls eine Addition angezeigt wird. Sa_1 wird dadurch aktiviert.

Sa_1 aktiviert Sb_1 . Sb_1 aktiviert Sa_2 .

Sa_2 aktiviert Sb_2 . Sb_2 aktiviert Sa_3 oder Lz_0 .

Sa_3 aktiviert Sb_3 . Sb_3 aktiviert Sa_4 oder Lz_0 .

Sa_4 aktiviert Sb_4 . Sb_4 aktiviert Sa_5 .

Sa_5 aktiviert Sb_5 . Sa_5 aktiviert Lz_0 .

Die Steuerung der Relaiskette erfolgt durch die Sa Relais. Durch die Aktivierung der verschiedenen Sa Relais in Abhängigkeit von Av , dem Vorzeichen der beiden Summanden, werden die verschiedenen E und F Steuerrelais der Additionswerke A und B gesteuert. Die Sb Relais sind reine Zwischenrelais, welche die Weiterschaltung der Relaiskette zum nächsten Sa Relais veranlassen.

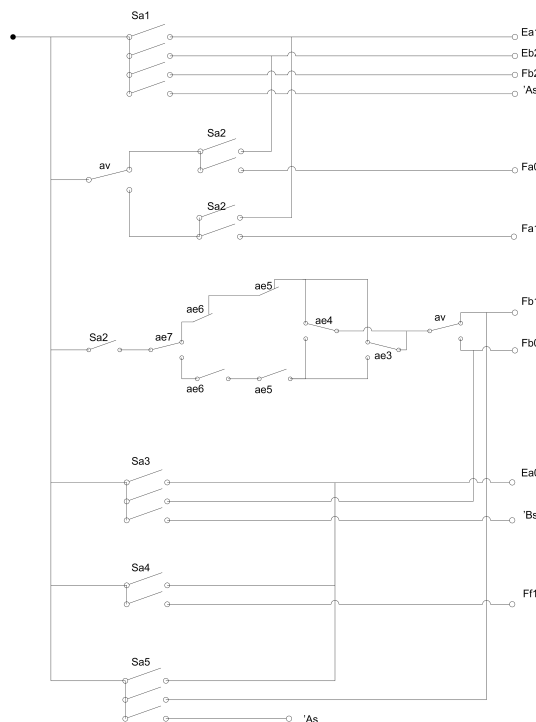


Abbildung 40: Schaltung der Addition

Im ersten Spiel wird nun Sa_1 geschaltet, was Ea_1 , Eb_2 , Fb_2 und $'As$ aktiviert. Ea_1 überträgt den Exponenten des 1. Operanden auf das Register Aa. Eb_2 überträgt

den Exponenten des 2. Operanden auf Ab und durch Fb_2 wird die Mantisse des 2. Operanden über die große Weichenstraße, ohne eine Verschiebung, auf Bb gegeben. Dadurch wird der Wert der Mantisse des 2. Operanden von Bb auf Be übertragen und kann von dort im 2. Spiel auf Ba oder Bb gegeben werden. Von Bg wäre eine Übertragung nach Ba nicht möglich. 'As aktiviert das Additionswerk A, welches die Differenz beider Exponenten bildet. Die Operanden befinden sich wie folgt auf den Registern:

Aa: 1. Operand	Bf: 1. Operand
Ab: 2. Operand	Bb: 2. Operand

Im 2. Spiel wird nun über Sa_2 die eigentliche Addition durchgeführt. In Abhängigkeit des Vorzeichens wird somit entweder Eb_2 , Fa_0 und Fb_1 aktiviert oder Ea_1 , Fa_1 und Fb_0 . Eb_2 leitet den Exponenten des 2. Operanden auf Ab. Fa_0 die Mantisse des 2. Operanden, „welche sich jetzt auf Be befindet“ auf Ba und Fb_1 den 1. Operanden Mantisse auf Bb. Die Mantisse des ersten Operanden ist nun auf Bb und die des 2. Operanden auf Ba. Bei anderem Vorzeichen aktiviert Ea_1 den Exponenten des 1. Operanden auf Aa. Fa_1 setzt die Mantisse des 1. Operanden auf Ba und Fb_0 setzt „den Wert des Be Registers“, also des 2. Operanden auf Bb. Es befindet sich somit der Exponent des 2. Operanden auf Ab und entweder die Mantisse von 1 auf Ba und die von 2 auf Bb oder umgedreht.

Ab: 2. Operand	Aa: 1. Operand
Ba: 2. Operand	Ba: 1. Operand
Bb: 1. Operand	Bb: 2. Operand

Auf Bb befindet sich der ursprünglich kleinere Wert, da dieser über die große Weichenstraße verschoben werden muss. Durch die im 1. Spiel berechnete Exponentendifferenz, konnte die Mantisse welche auf Bb übertragen wird, um die entsprechende Anzahl an Stelle verschoben werden. Dadurch kann nun in Abhängigkeit der Operation und der Vorzeichen eine Addition oder Subtraktion durchgeführt werden.

In Spiel 2 wird nun Sa_2 aktiviert, welches Sb_2 aktiviert, welches in verschiedenen Abhängigkeiten durch Vg und Bt die Relais Sa_3 , Sa_4 oder Lz_0 schalten kann. Wurde eine Addition durchgeführt und durch Vg angezeigt, dass beide Operanden das gleiche Vorzeichen besitzen, die Addition also positiv ist, spricht das Relais Br an

und führt ein Ausrichten der Mantisse durch. Bei ungleichen Vorzeichen spricht das Relais Bs an und führt die Komplementbildung durch. Damit ist die Zahl negativ. Handelt es sich um das gleiche Vorzeichen wird lz_0 zum Ende der Operation geschaltet. Es muss nun nur noch durch die Aktivierung von Br die Mantisse ausgerichtet werden.

Handelt es sich um eine Subtraktion und unterschiedliche Vorzeichen sprechen Vg und Bt nicht an. Bt ist direkt mit dem Register Be verbunden und spricht bei positivem Ergebnis an. In diesem Fall wird Sa_3 wie folgt geschaltet. Mit Sa_3 wird Ea_0 , $'Bs$ und Fb_0 aktiviert. Ea_0 führt zu einem blindem Umlauf im Additionswerk A. Fb_0 legt das Be Ergebnis über die große Weichenstraße auf Bb. $'Bs$ führt zur Komplementbildung der Mantisse, da das Ergebnis negativ ist.

Es wurde eine Subtraktion mit positivem Ergebnis durchgeführt. Allerdings entspricht die Mantisse nicht der Normalform, sonder der Wert ist $\leq 1,0$. Es wird Sa_4 aktiviert mit Ea_0 einen weiteren blinden Umlauf im Additionswerk A. Ff_1 legt das Ergebnis der Operation auf Bf. Dieses muss allerdings noch ausgerichtet werden.

Sa_5 aktiviert mit Ea_0 einen weiteren blinden Umlauf im Additionswerk A. Fb_1 legt das Ergebnis, welches gerade auf Bf übertragen wurde, auf Bb. Die Verschiebung erfolgt somit um die Stellen, um die die Mantisse verschoben werden muss. $'As$ führt eine weitere Addition im Additionswerk A durch und passt damit Alpha an.

Eine Addition dauert drei Spiele und damit ca. 0,6 Sekunden. Eine Subtraktion ist in vier oder fünf Spielen möglich und dauert damit ca. eine Sekunde.

6.2 Multiplikation

Die Multiplikation wird durch eine Aneinanderreihung von Additionen realisiert. Diese ist ebenfalls durch eine Relaiskette ausgeführt. Der Multiplikator befindet sich als 1. Operand auf Bf und der Multiplikand als 2. Operand auf Bg.

Auf den Stellen der Mantisse, auf denen der Multiplikator eine Null hat, wird keine Addition durchgeführt, auf denen er eine Eins hat schon. Der Multiplikand wird jeweils von Bg über die große Weichenstrasse auf Bb gegeben.

Wie bei allen Rechenoperationen, muss auch hier am Ende Br zum Ausrichten der Mantisse geschaltet werden. Dadurch wird dann auch Lz aktiviert und so das Ende der Rechnung angezeigt.

Bei der Multiplikation muss durch alle Stellen der Mantisse, maximal 24-mal durch-

gegangen werden. Bei fünf Takten pro Sekunde dauert eine Multiplikation damit knapp fünf Sekunden. Wird eine gewisse Ungenauigkeit in Kauf genommen werden, welche auf 3-4 Dezimalstellen beschränkt ist, lässt sich die Dauer der Multiplikation entsprechend auf ca. drei Sekunden verkürzen.

6.3 Division, Quadratwurzel, Multiplikation mit Pi

Die Operationen der Division, des Quadratwurzelziehens und der Multiplikation mit Pi sind etwas aufwändiger, als die gerade gezeigte Multiplikation- und Additionsverfahren. Diese Schaltungen führen ihre Teiloperationen gesteuert durch Schrittschalter durch. Die Division wird durch Schrittschalter Id, Quadratwurzel durch Schrittschalter Wd und Multiplikation durch Schrittschalter Hd gesteuert.

Der Schrittschalter Id wird einfach über das Relais Li , den Befehl der Division angesteuert. Das Wurzelziehen wird durch das Relais Lw_1 kommandiert und setzt voraus, dass ix nicht aktiviert ist. Aus einer imaginären Zahl kann keine Wurzel gezogen werden. Die Multiplikation mit Pi erwartet Lm_1 , $\overline{Pb_0}$ und $\overline{Pb_1}$ um den Schrittschalter Hd zu aktivieren. Voraussetzungen für alle drei Operationen, ist, dass $\overline{S_x}$ und $\overline{S_y}$ gilt. Die Operationen können nur durchgeführt werden, wenn es sich um keine Sonderwerte handelt.

6.3.1 Division

Die Division wird durch den Schrittschalter Id gesteuert, welcher den Divisor vom Rest des Dividenten addiert und subtrahiert, in Abhängigkeit des Vorzeichens des vorhergehenden Rest. Für die Division sind 26 Spiele notwendig. In Spiel 1 wird Ia aktiviert, in Spiel 2-24 Ib und Ic in Spiel 25. Spiel 26 zeigt das Ende der Operation mit der Aktivierung von Lz_0 an. Gleichzeitig wird in Spiel 2, Ft geschaltet und in den Spielen 3-25 die Relais Bf_{-1} bis Bf_{-23} .

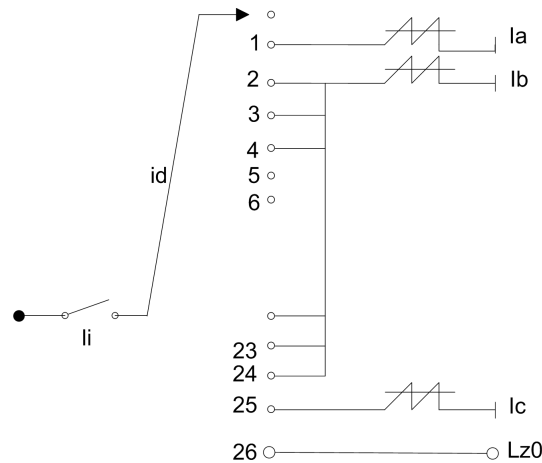


Abbildung 41: Schrittschalter Division

Ia schaltet Ea_1 , Eb_2 , Fa_1 , Fb_2 , $'Bs$, $'As$. Hiermit werden die Operanden aus den Af, Ag Registern in Aa, Ab übertragen, sowie Bf, Bg auf Ba, Bb. Die Befehle $'Bs$, $'As$ bilden die jeweilige Differenz der Werte von Exponent und Mantisse.

Ib schaltet Fa_1 , Ea_0 , Fa_0 , Fq , Fb_2 , $'Bs$. Durch Ea_0 erfolgt jeweils ein blinder Umlauf des Exponenten durch das Additionswerk A. Durch Fa_0 und Fq wird die soeben berechnete Differenz um jeweils einen Wert aufwärts verschoben. Über Fb_2 wird der Divisor in jedem Takt auf Bb übertragen und in Abhängigkeit des Vorzeichens des vorhergehenden Restes addiert oder subtrahiert.

Gleichzeitig aktiviert der Schrittschalter, in den Schritten 3-25, die Relais Bf_{-1} bis Bf_{-23} und erzeugt so auf diesen das Ergebnis der Division. Auf den jeweiligen Bf Stellen wird eine 1 oder 0 gesetzt. Eine 1 wird gesetzt wenn der neue Rest positiv ist. Eine 0 wird gesetzt wenn der neue Rest negativ ist. Bf_0 wird über Ft belegt, da zu diesem Zeitpunkt der Dividend noch nicht gelöscht ist.

Ic schaltet in Spiel 25 die Register Ea_0 , Fa_1 , Fq , $'As$ und Ab_0 werden in Abhängigkeit von Bf_0 geschaltet. Befindet sich auf Bf_0 kein Wert, hat die Mantisse einen Wert von kleiner 1,0. Dieses ist nicht zulässig, weshalb durch Fa_1 und Fq eine Verschiebung der Mantisse über die kleine Weichenstraße auf Ba erfolgt. Gleichzeitig wird durch die Addition von eins, der Exponent um eins verringert. Dieses wird erreicht, indem der Exponent im Register Aa und eine Konstante auf Ab gegeben wird. Eine einfache Subtraktion verringert den Exponenten.

Eine Division ist damit in 26 Spielen und ca. 5,2 Sekunden möglich.

6.3.2 Quadratwurzel

Die Schaltung des Quadratwurzelziehens ist der der Division sehr ähnlich. Allerdings wird das Resultat vom Radikand abgezogen. In Spiel 1 wird Wa aktiviert. In den Spielen 2-24 Wb und Ic in Spiel 25. Das Ende wird in Spiel 26 durch Lz_0 angezeigt. Wa aktiviert Ea_2 und Fa_1 . Fq wird zusätzlich in Abhängigkeit von Af_0 geschaltet. Durch Ea_2 wird der Exponent um eine Stelle nach unten verschoben und so halbiert. Fa_1 bringt den Wert der Mantisse auf das Register Ba. Ist der Exponent ungerade wird dieses durch Af_0 angezeigt und aktiviert dann Fq , welches die Mantisse beim Verschieben um eine Stelle aufwärts verschiebt.

In den Schritten 2-24 wird analog zur Division Wb aktiviert, welches Ea_0 , Fa_0 , Fb_1 , Fq und Fn aktiviert. Parallel dazu wird wieder Bf_{-1} bis Bf_{-23} in den Spielen 3-25 angesprochen. Hierdurch wird der Rest des Radikand in jedem Spiel um eine Stelle aufwärts verschoben. Die Bildung des Resultats auf Bf entspricht der der Division. Wie bei der Division, wird in jedem Spiel entschieden, ob die Belegung der Bf Relais mit einer 0 oder 1 erfolgt. Die Annahmen hierfür sind mit denen der Division identisch.

Die Operation der Quadratwurzel ist damit in ebenfalls 26 Spielen und ca. 5,2 Sekunden möglich.

6.3.3 Multiplikation mit Pi

Die Multiplikation mit Pi erfolgt durch den Schrittschalter hd_1 . Die Darstellung der Zahl Pi im Sekundalsystem sieht wie folgt aus: $2^2 * 1,1001001000011111101101$. Gibt es nun eine Multiplikation mit Pi, muss auf jeder Stelle auf der die Zahl Pi im Dualsystem eine 1 besitzt, eine entsprechende Addition, wie bei der reinen Multiplikation, durchgeführt werden. Die Zahl Pi wird bei der Multiplikation als 2. Operand betrachtet. Da allerdings schon vor ausführen der Multiplikation feststeht, auf welchen Stellen die Zahl Pi eine eins besitzt und damit eine Addition durchgeführt werden muss, kann dieses auch durch eine feste Verdrahtung der Schaltungen erfolgen. Die Multiplikation ist dadurch schneller durchführbar. Durch die Feste Verdrahtung der Zahl Pi, kann diese durch die Fw Relais dargestellt werden. Die Verschiebung wird durch die Fw Relais erreicht. Die Schaltung aktiviert in Schritt 1-13 Ha. Ha schaltet Fa_0 , Fb_1 und Eb_0 . Gleichzeitig in Schritt 12 Ea_1 und in Schritt 13 Hb.

In den Schritten 1-13 werden die Fw geschaltet, welche eine Verschiebung veranlassen. In der Reihenfolge von Schritt 1-13 werden die Relais Fw_0 , Fw_1 , Fw_4 , Fw_7 , Fw_{12} , Fw_{13} , Fw_{14} , Fw_{15} , Fw_{16} , Fw_{17} , Fw_{19} , Fw_{20} , Fw_{22} aktiviert. In Schritt 13 wird durch Hb noch Br zum Ausrichten und Aa_0 geschaltet. In Schritt 14 wird Lz_0 zum Beenden der Operation angezeigt. Die Multiplikation mit Pi ist somit in 14 Spielen durchführbar und benötigt knapp 3 Sekunden.

6.4 Hilfsoperation $\times 1/2$, $\times 2$, $\times 10$

Die Steuerung der Hilfsoperationen wird wiederum durch eine Relaiskette gesteuert. Die Operationsrelais Lh2 bzw. Lh3 lösen eine Relaiskette Hc, Hd, Lz_0 aus, welche die nötigen Steuerungen bewirken.

Die Hilfsoperationen $\times 1/2$, $\times 2$, $\times 10$ sind einfache Operationen und daher durch eine Relaiskette ausgeführt. Lh_2 startet die Multiplikation mit $1/2$ oder multipliziert mit zwei. Lh_3 startet die Division oder Multiplikation mit zehn. Die Relaiskette der Form Hc, Hd, Lz_0 führt die notwendigen Steuerungen durch.

Hc aktiviert Ea_1 und Fa_1 , wodurch der erste Operand in das Register Aa für den Exponenten und Ba für die Mantisse übertragen wird. Gleichzeitig wird der Wert in Register Ab_0 auf eins gesetzt. In Abhängigkeit von Lh_2 und Pb'_0 , was der Multiplikation mit zwei entspricht wird $'As$ geschaltet und damit der Exponent um eins erhöht. Die Zahl ist mit zwei multipliziert. Bei der Division durch zwei reicht es das Steuerrelais Ea_2 zu aktivieren, was einer Abwärtsverschiebung des Exponenten von einer Stelle und damit der Division durch zwei entspricht.

Die Multiplikation mit zehn wurde bereits bei der Eingabe von Dezimalzahlen behandelt. Die Addition der Verschiebung der Mantisse um eine Stelle und der Verschiebung um drei Stellen ergibt den Wert der Multiplikation mit zehn.

Damit lassen sich die Hilfsoperationen in maximal fünf Takten und damit ebenfalls unter einer Sekunde durchführen.

6.5 x^2 und $1/x$

Die Operationen des Quadrierens oder des Kehrwertes können sehr einfach durch eine Relaiskette realisiert werden. Die Operationen Multiplikation und Division wur-

den bereits implementiert. Daher muss nur noch der Wert des ersten Operanden von Af auf Ag und von Bf nach Bg übertragen werden. Eine direkte Übertragung ist nicht möglich, weshalb eine Übertragung über die Register der Additionswerke erfolgen muss. Der Operationsbefehl aktiviert Lh_5 welches Lh_6 und dann Lh_7 aktiviert. In Abhängigkeit von Pb'_0 wird entweder Lm_0 für die Multiplikation oder Li für die Division geschaltet. Lh_6 aktiviert die Steuerrelais des Rechenwerks Eb_1 , Eb_0 , Eg_1 , Fb_1 , Fb_0 , Fg_1 .

Eb_1 gibt den Exponenten von Af auf Ab. Eg_1 den Wert auf Ag. Fb_1 den Wert der Mantisse von Bf auf Bb. Fg_1 den Wert der Mantisse auf Bg. Die Operanden befinden sich nun auf beiden Operandenregistern im Additionswerk. Für die Quadrierung muss jetzt nur der Befehl Lm_0 gegeben werden und beide Werte werden miteinander multipliziert.

Soll der Kehrwert gebildet werden, muss noch auf Bf der Wert eins eingestellt werden, damit dann die Mantisse des 1. Operanden (1) durch den 2. Operanden dividiert werden kann. Ausgeführt durch Li , was die Division aktiviert. Die eins wird durch Lh_2 und Pb'_0 auf Bf_0 gegeben. Damit ist dann auch der Kehrwert einer Zahl gebildet. Durch die maximal zusätzlichen drei Takte und der Division oder Multiplikation, ist das Quadrieren und Bilden des Kehrwertes in unter 30 Takten und damit unter 6 Sekunden möglich.

6.6 Sonderwerte

Teilschaltung N zeigt die Behandlung der Sonderwerte im Rechenwerk. Hierbei wird insbesondere auf die Ein- und Ausgabe sowie das Speichern dieser Wertigkeiten eingegangen. Sonderwerte werden im Ganzen durch die Belegung der Register I (Imaginär), V (Vorzeichen) und S (Sonderwerte, sehr klein, sehr groß) angezeigt. Die Indizes x sind für den ersten Operanden, also I_x , V_x , S_x , die des zweiten durch I_y , V_y , S_y gekennzeichnet.

Die Sonderwerte können über die Kontakte t101, t108 und t129 von der Tastatur übertragen werden. Im Speicher sind die Sonderwerte auf den ersten drei Bit gespeichert und werden über die Speicherleitungen c_1 , c_2 und c_3 übertragen. Die Sonderwerte, welche über die Tastatur und damit den Ta Relais übertragen werden, können nur die Register I_x, V_x, S_x aktivieren. Die von der Speicherzelle durch Steue-

rung von Ef_2 auf X oder Eg_2 auf Y. Das Ende einer Rechenoperation wird stets mit der Aktivierung des Relais Lz_0 beendet, welches in Abhängigkeit der Relais V_r Vorzeichen, I_r Imaginär und S_r Sonderwerte die Relais I_r , V_r , S_r belegt. Voraussetzung ist hierfür die Aktivierung von Ea_0 welche den blinden Umlauf im Additionswerk veranlasst. Ist eine Rechnung beendet und das Ergebnis von Ae durch Ef_1 auf Af gegeben, so werden auch die X Sonderwerte durch V_r , I_r und S_r belegt. Wird das Ergebnis durch Eg_1 auf den 2. Operanden Ag gegeben, wird gleichzeitig I_y, V_y, S_y belegt. Soll ein Wert von Ae auf das Speicherwerk übertragen werden, was durch Fc veranlasst wird, wird dadurch Ca_{1-3} für die Sonderwerte belegt.

Über die Schaltung von I_y, V_y, S_y können die Sonderwerte für die Ausgabe auf die Leitungen t201, t208 und t229 gegeben werden können. Diese werden dann entsprechend vom Lampenfeld angezeigt. Auffällig ist, dass I_y und S_y bei dem Vorliegen eines Imaginär oder Sonderwertes anspringen, V_g aber angezogen ist, wenn es sich um einen positiven Wert handelt. Positive Werte werden somit explizit angezeigt.

Ist eine Rechnung beendet und liegt ein Sonderwert beim ersten Operanden vor, werden die aktivierten I_x, V_x, S_x Relais auf Y übertragen, wenn die Zahl ausgegeben werden soll. Eine Ausgabe ist nur vom 2. Operanden, den Ag, Bg Relais und daher auch nur über den Index Y möglich. Wird allerdings eine weitere Zahl von der Eingabe der Tastatur erwartet, durch Lu angezeigt, muss die gerade auf dem 1. Operanden befindliche Zahl ins Speicherwerk übertragen werden. Ist das Speicherwerk noch nicht für die Aufnahme eines Wertes bereit, werden die Sonderwerte in den I_r, V_r, S_r Relais zwischengespeichert.

7 Fazit

Die Analyse der Z4 zeigt deutlich, dass Zuse es vermieden hat, beim letzten geplanten Prototypen vor der Serienfertigung große Änderungen im logischen Aufbau vorzunehmen. In erster Linie sind die Veränderungen zum Vorgängermodell der Steigerung der Bedienfreundlichkeit geschuldet. Der Einsatz des Lampenfeldes verringert Fehleingaben von falschen Variablen. Der vergrößerte Befehlssatz ermöglicht es einfachere Rechenpläne zu erstellen und insgesamt komplexere Rechnungen mit weniger Aufwand durchzuführen. Durch die vergrößerte Mantisse sind die Zahlenräume verkleinert und die Genauigkeit der Maschine verbessert worden. Auch wenn die Taktrate von 5Hz in etwa dem des Vorgänger Modells entspricht, wurde die Z4 mehrere Jahre in Betrieb gehalten. Die Zuverlässigkeit und Wartungsarmut ermöglichte eine Einsatzzeit von mehr als acht Jahren[5].

Die Änderungen nach Kriegsende, die für den Verkauf der Z4 an die ETH Zürich notwendig waren, ließen sich verhältnismäßig einfach realisieren und unterstreichen damit den Erfolg des modularen Aufbaus. Im Wesentlichen bestand die Genialität von Zuse in der Konstruktion der logischen Schaltungen. Diese sind zum Teil in abgewandelter Form bis heute im Einsatz (Carry-Look-Ahead Addierer) und sind nicht an die Relais-technik gebunden. Mit der Wahl anderer Bauteile wie den Röhrenrelais, wäre die Rechengeschwindigkeit bis zu 1000-mal höher. Große Änderungen an den Schaltungen wären nicht notwendig gewesen.

Die Rechenmaschine Z4 war für das 20. Jahrhundert wegweisend und hat den Grundstein für die heutige PC Architektur gelegt. Die von Zuse verwendete Blockarchitektur findet bis auf kleinere Änderungen bis heute Anwendung. Die Aufteilung in Rechenwerk, Planwerk, Kommando- und Ablesevorrichtung sind die nächsten Jahre nicht verändert worden. Insbesondere die Trennung von Programm- und Zahlenwertspeicher ist zum damaligen Zeitpunkt einzigartig gewesen.

Als einer der ersten verwendete Zuse das Dualsystem[11], in halblogarithmischer Form. Der Einsatz des gleitenden Kommas, hat die Z4 wie auch die Vorgängermodelle universell einsetzbar gemacht. Der Programmierer musste das mögliche Ergebnis einer Operation nicht mehr vorher abschätzen. Die Gefahr eines Unter- oder Überlauf war deutlich geringer, als bei anderen Maschinen in dieser Zeit, welche ein festes Komma verwendet haben.

Wesentlich war die hardwareseitige Implementierung und Überwachung der Ergeb-

nisse auf Sonderwerte. Die Z4 war zum damaligen Zeitpunkt die einzige Rechenmaschine, die Sonderwerte durch Hardwareimplementierungen abfing. Dieses hat dem Programmierer viel Arbeit abgenommen, unnötige Rechnungen mit Sonderwerten vermieden und so die Effizienz der Z4 gesteigert.

Literatur

- [1] Konrad Zuse: „Schaltzeichnungen für das Gerät Z4“
ZuP 003/002, Berlin, 1944
- [2] Konrad Zuse: „Notizen zu den Schaltungen für das Gerät Z4“
ZuP 003/003, Berlin, 1944
- [3] Konrad Zuse: „Beschreibung des Gerätes Z4. Korrespondenz mit DVL.
Anlage zum Schreiben vom 28.9.1941“
ZuP 021/001, Berlin, 1941
- [4] Konrad Zuse: „The Computer - My Life“
Berlin: Springer-Verlag, 1993
- [5] Eduard Stiefel: „Rechenautomaten im Dienste der Technik. Erfahrungen
mit dem Zuse-Rechenautomaten Z4. Diskussion “
ZuP 041/002, Zürich
- [6] Konrad Zuse: „Entwicklungslinien einer Rechengeräte- Entwicklung von
der Mechanik zur Elektronik. “
ZuP 040/006, Bad Hersfeld, Deutschland 1960
- [7] Raúl Rojas: „Die Architektur der Rechenmaschinen Z1 und Z3“.
August 2000
- [8] Konrad Zuse: „Patentschrift Nr. 924107“.
ZuP 005/002, Deutschland 1953
- [9] Ursula Schweier, Dietmar Saupe: „Funktions- und Konstruktionsprinzipi-
en der Programmgesteuerten mechanischen Rechenmaschine Z1 “.
Gesellschaft für Mathematik und Datenverarbeitung mbH , 1988
- [10] Konrad Zuse. „Bericht über meine Geräte“.
ZuP 010/010, 1945
- [11] Konrad Zuse: „Ein Neues Rechengerät für technische und wissenschaftliche
Rechnungen“.
ZuP 037/007, Hopferau 1948

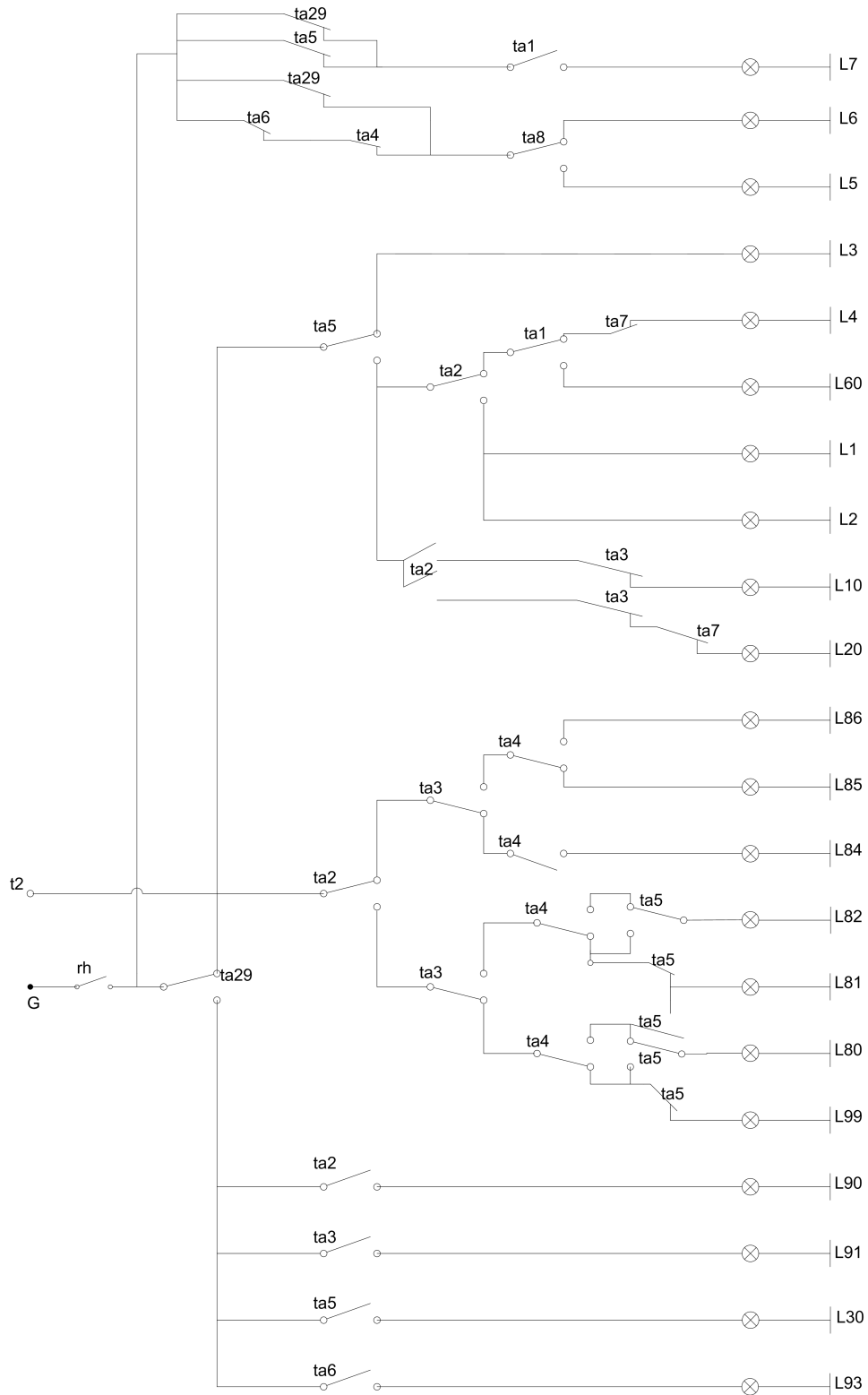
- [12] Konrad Zuse: „*Datenblatt der Z4*“.
ZuP 037/013, 1950
- [13] Konrad Zuse: „*Rechenpläne für das Rechengerät V 4 Herstellungsanweisung und Musterbeispiele*“.
ZuP 037/014, 1952
- [14] Konrad Zuse: „*Der Programmator*“.
ZuP 039/10, Neukirchen , 1952
- [15] Raúl Rojas: „*Z1, Z2, Z3 and Z4*“.
1999
- [16] Konrad Zuse: „*Rechenvorrichtung aus mechanischen Schaltgliedern*“.
ZuP 019/003, Berlin: Springer-Verlag, 1952
- [17] Konrad Zuse: „*Theorie der mechanischen Schaltglieder*“.
ZuP 019/001, Berlin: Springer-Verlag, 1952
- [18] Raúl Rojas: „*Konrad Zuse (1910-1995) Kurzbiographie von Raúl Rojas*“.
- [19] Raúl Rojas: „*How to Make Zuse's Z3 a Universal Computer*“.
- [20] Raúl Rojas, Alexander Thurm: „*Patentanmeldung Z391*“.
ZuP 005/011, 1941
- [21] Hartmut Petzold: „*Moderne Rechenkünstler - Die Industrialisierung der Rechentechnik in Deutschland Kapitel 5: Ein eigensinniger Erfinder-Unternehmer: Konrad Zuse*“.
Verlag C.H. Beck München, 1992
- [22] Konrad Zuse: „*Planfertigungsgeräte*“.
ZuP 010/024, Berlin: Springer-Verlag, 1944
- [23] „*Kriegsauftrag (1945)*“.
ZuP 037/016, 1945
- [24] Konrad Zuse: „*Rechenplangesteuerte Rechengeräte für technische und wissenschaftliche Rechnungen*“.
ZuP 009/006, 1943

- [25] „Die Rechenmaschine des Ingenieurs (1938) Mathematische Probleme “.
ZuP 009/003, Berlin, 1938
- [26] Konrad Zuse: „Einführung in die allgemeine Dyadik“.
ZuP 009/004, 1938
- [27] Konrad Zuse: „Die Rechenmaschinen des Ingenieurs “.
ZuP 009/001, 1936
- [28] Konrad Zuse: „Unterlagen-Zusammenstellung zur Z4 “.
1941

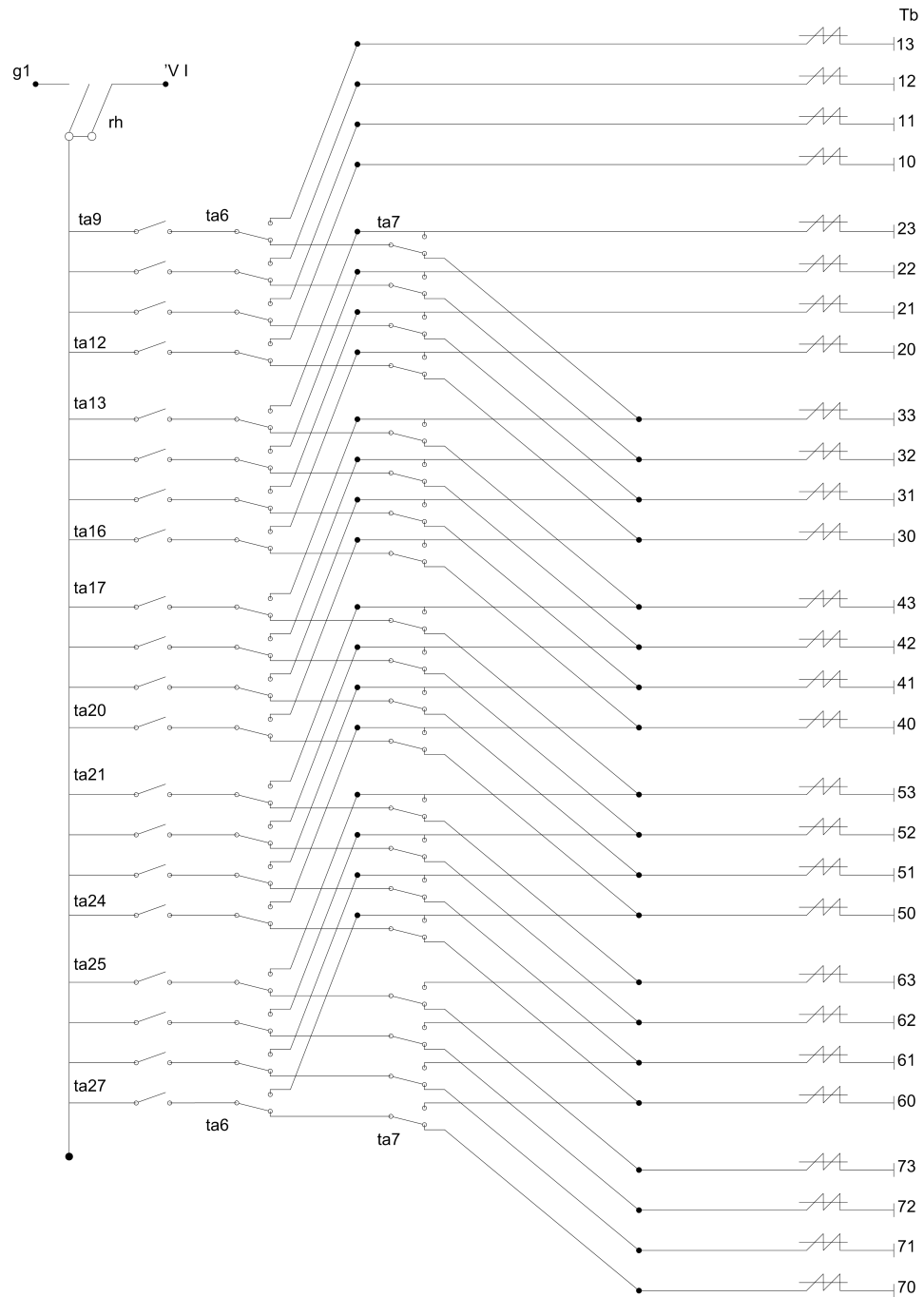
A Anhang

A.1 Lampenfeldanzeige

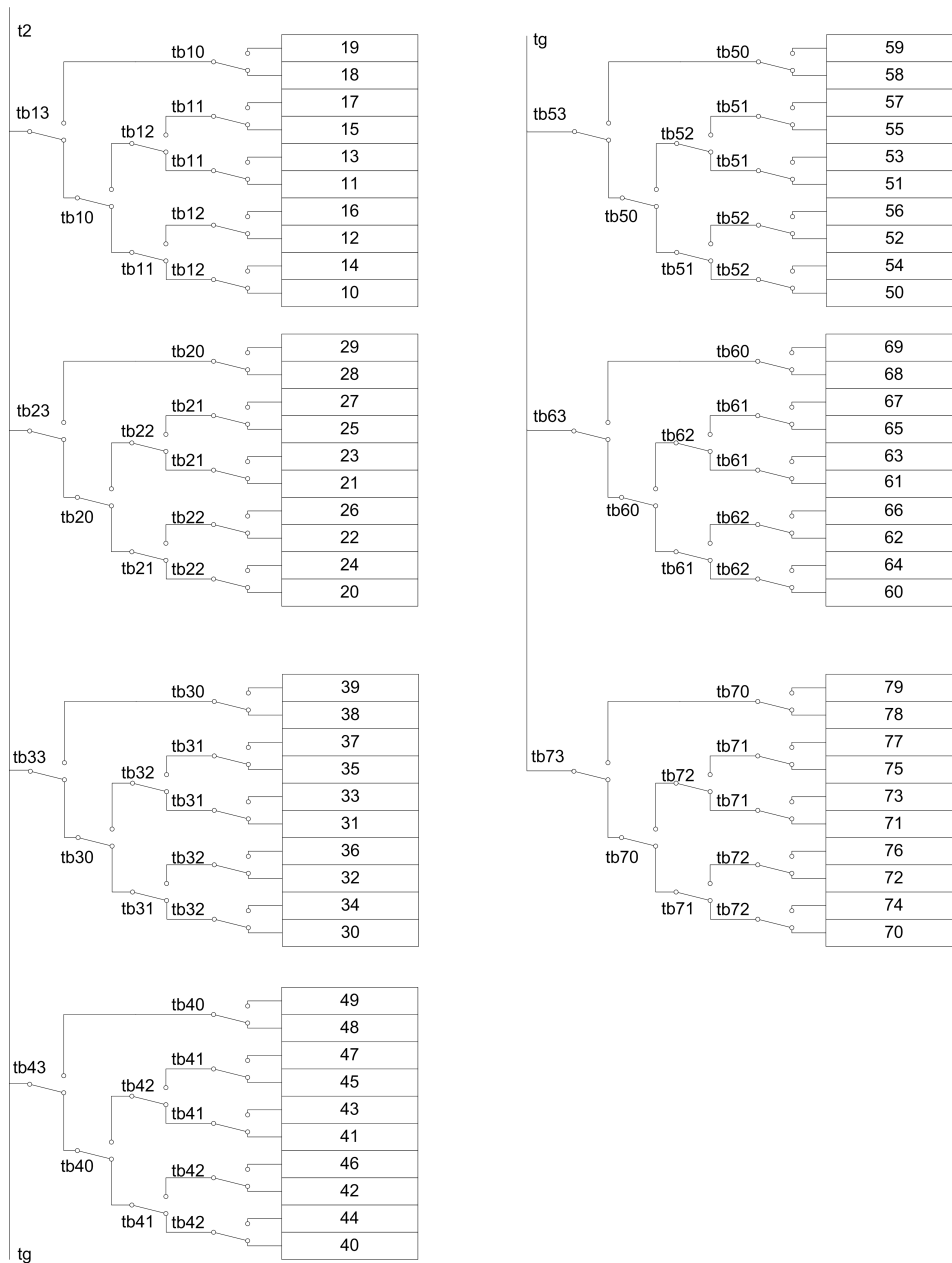
A.1.1 Lampenfeldanzeige durch Ta Relais



A.1.2 Weichenstraße Ta auf Tb Relais

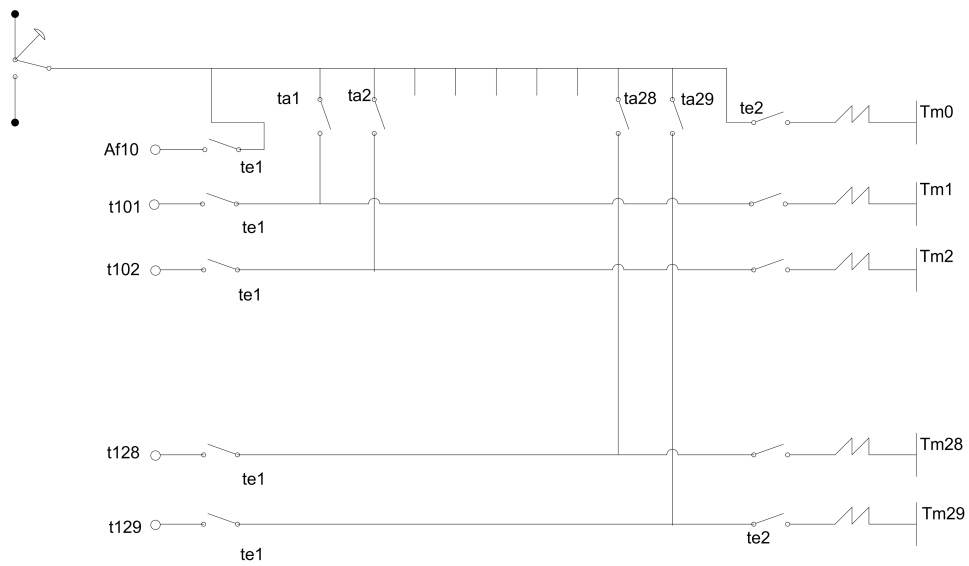


A.1.3 Lampenfeldanzeige durch Tb Relais

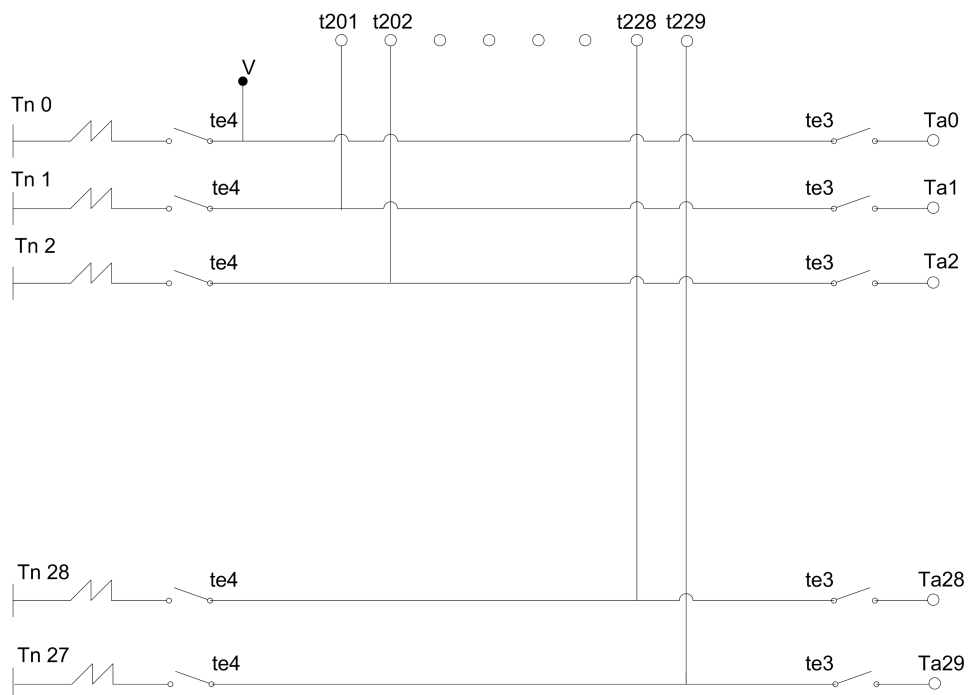


A.2 Tastaturrelais

A.2.1 Ta Relais auf Rechenwerk / Tm Relais

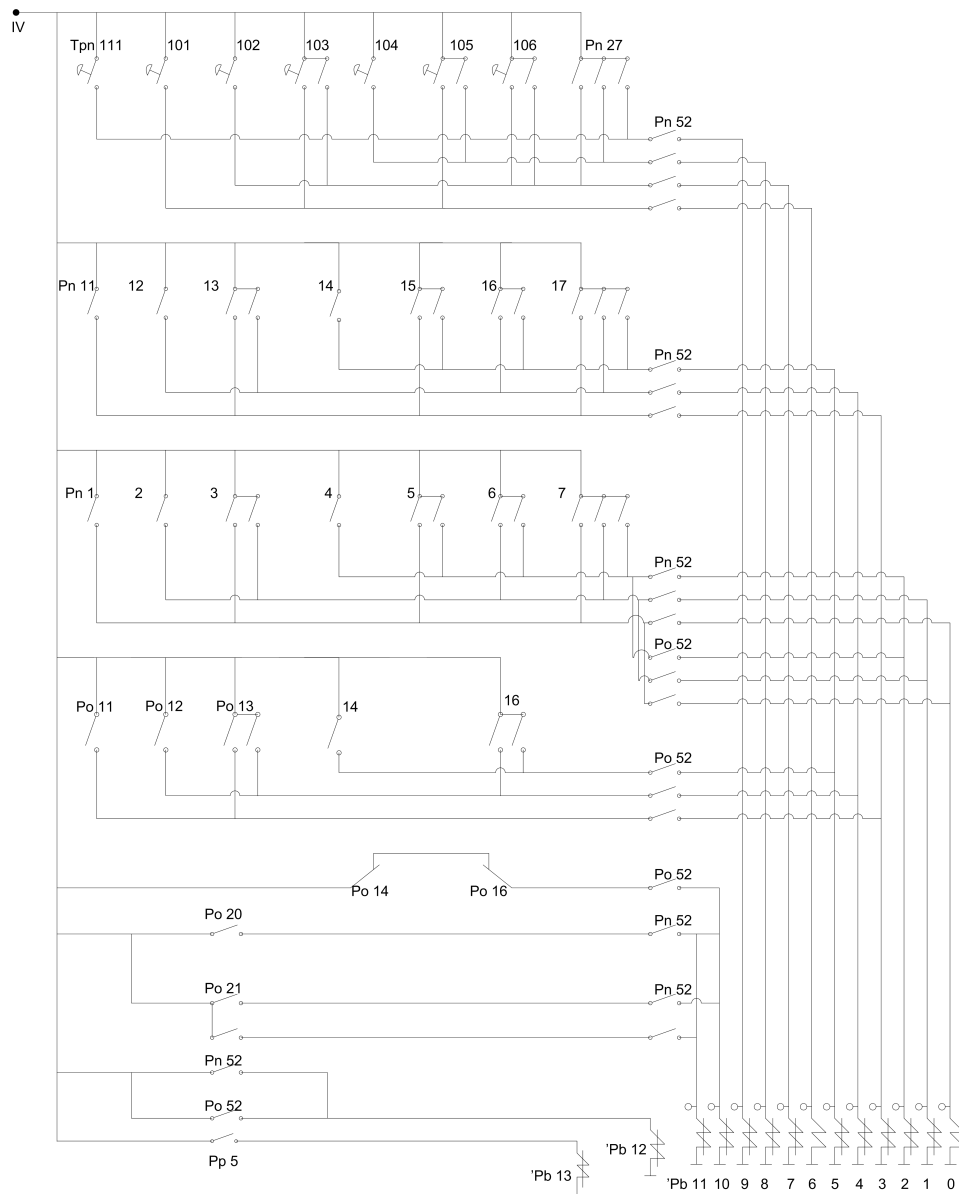


A.2.2 Rechenwerk auf Tn / Ta Relais

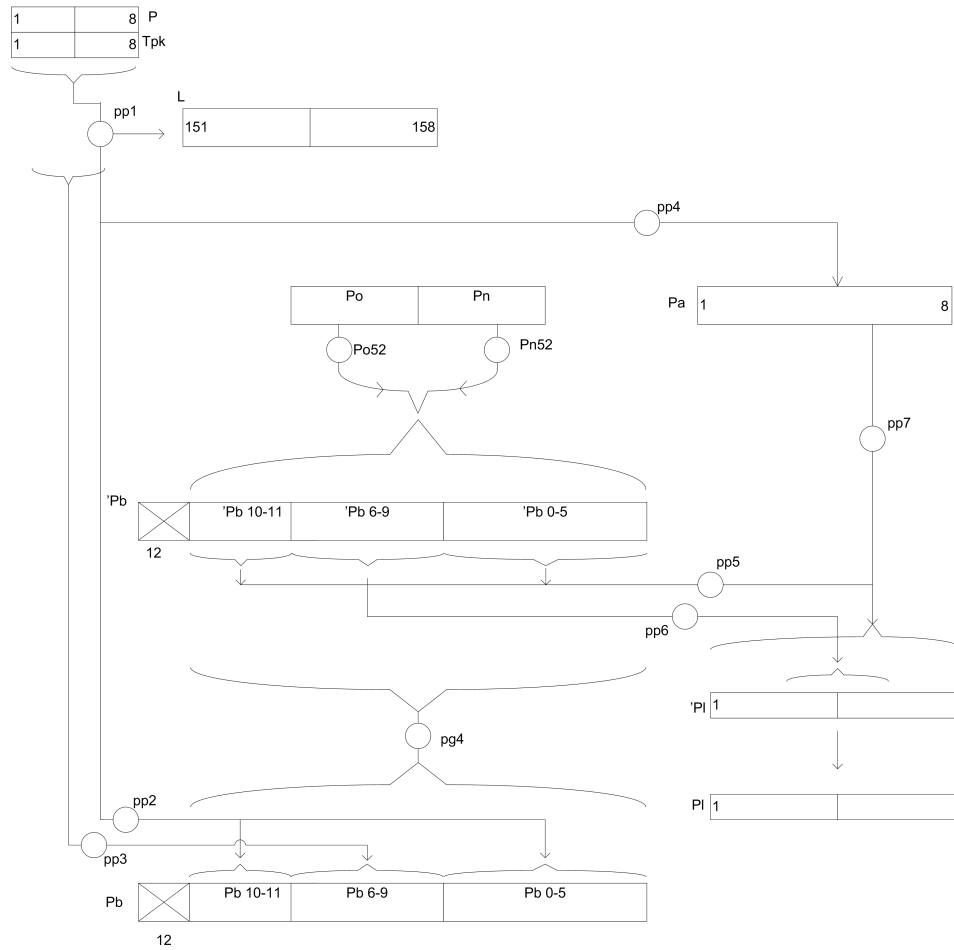


A.3 Planwerkschaltungen

A.3.1 Belegung der Pb Relais durch Tastatureingabe



A.3.2 Befehlskreislauf Planwerk



Erklärung

Hiermit versichere ich, die vorliegende Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben und an allen Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, diese als solche kenntlich gemacht habe. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt.

Berlin, den 15. März 2010 David Anders