

Stefan Höltgen

Hardwhere – Softwhere. Eine Archäologie der/als Navigation durch Strukturspeicher

2022

<https://doi.org/10.25969/mediarep/18792>

Veröffentlichungsversion / published version

Zeitschriftenartikel / journal article

Empfohlene Zitierung / Suggested Citation:

Höltgen, Stefan: Hardwhere – Softwhere. Eine Archäologie der/als Navigation durch Strukturspeicher.
In: *Navigationen - Zeitschrift für Medien- und Kulturwissenschaften*, Jg. 22 (2022), Nr. 1, S. 139–
161. DOI: <https://doi.org/10.25969/mediarep/18792>.

Nutzungsbedingungen:

Dieser Text wird unter einer Creative Commons -
Namensnennung - Weitergabe unter gleichen Bedingungen 4.0/
Lizenz zur Verfügung gestellt. Nähere Auskünfte zu dieser Lizenz
finden Sie hier:

<https://creativecommons.org/licenses/by-sa/4.0/>

Terms of use:

This document is made available under a creative commons -
Attribution - Share Alike 4.0/ License. For more information see:

<https://creativecommons.org/licenses/by-sa/4.0/>

HARDWHERE – SOFTWHERE

Eine Archäologie der/als Navigation durch Strukturspeicher

VON STEFAN HÖLTGEN

ABSTRACT

Speichertechnologien, die Informationen über Zeiträume sichern, stellen nur eine Form der Speicherung in Medien dar. Die Architektur und Struktur von Medientechnologien selbst kann als Strukturspeicher betrachtet werden; sie besitzt eigene Möglichkeiten der Sicherung und benötigt spezifische Techniken der Bewahrung. Der Beitrag versucht solche Strukturspeicher zu definieren, indem der Begriff historisch, technologisch und linguistisch hergeleitet wird. Im zweiten Teil werden verschiedene Strukturspeicher diskutiert und Formen ihrer operativen Preservation vorgestellt. Insgesamt werden dabei Fragen verhandelt, wie sie auch für das Navigieren essenziell sind, insofern auf Wege, Orte, Adressen usw. fokussiert wird – allerdings nicht im geographischen Makro-, sondern technologischen Mikroraum.

KEYWORDS: Speicher, Struktur, Medienarchäologie, Informatik, CPU

»Ganz offensichtlich wird die Information
im Rechner in struktureller Weise gespeichert.«
(Heinz von Foerster)

I. EINLEITUNG

Von der Vielfalt vergangener Speichertechnologien sind heute, folgt man dem Speichertheoretiker Horst Völz, nur noch vier Arten übrig geblieben: die Magnetbänder, die Festplatten, DRAM-Speicher und Flashspeicher.¹ Mit der enormen Vergünstigung, Verkleinerung und Verbesserung (in Hinblick auf Haltbarkeit) der Flashspeicher, geraten die ersten drei Speicherarten weiter ins Hintertreffen – was sich im Personalcomputerbereich bereits deutlich zeigt. Datenspeicher, wie die genannten, bilden die technischen Realisationen jenes scheinbar scharf abgegrenzten Pols *Speicherung* der Medienfunktionstrias. Es scheint jedoch einiges dafür zu sprechen, dass eine kategoriale Abgrenzung dieser Funktionen bei Annäherung des

¹ Nicht vergessen werden dürfen hierbei natürlich die papiernen Speicher, auf denen in Form von Scribbles, Zeitschriften und Daten- und Programm listings immer schon gespeichert wurde.

(technischen wie historischen) Blicks aufweicht. Dies stellt zwar nicht die *Makrofunktionen* der *Verarbeitung*, *Übertragung* und *Speicherung*² von Informationen infrage, zeigt allerdings auf der Mikroebene Formen neuer Medienfunktionen und neue Blickweisen auf alte Technologien, bei denen letztlich »alles«³ als Speicher gesehen werden kann.

Der nachfolgende Beitrag möchte diese Überlegung einerseits im Begriff des Strukturspeichers *theoretisch* fundieren und dazu Konzepte von Verarbeitung, Speicherung und Übertragung an linguistische und diagrammatische Konzepte von Struktur(ierung), Ort und Weg koppeln. Zum anderen sollen diese Überlegungen an konkreten historischen und kontemporären Beispielen *praktisch dargestellt* werden. Im Zentrum steht dabei der LSI-Baustein 4004, den die Firma Intel am 15. November 1971 auf den Markt gebracht hat und der als der erste Mikroprozessor der Technikgeschichte 2021 seinen *50. Geburtstag* feierte. Der Intel 4004 wird aus diesem Anlass als ein spezifischer Strukturspeicher herausgegriffen; zuvor werden andere technische Strukturspeicher (Elektronikbaukästen, Murmelcomputer, mechanische Computer) vorgestellt, welche verschiedene Eigenschaften struktureller Speicherung realisieren. Die Frage, wie sich Speicherstrukturen selbst speichern (also über Zeiträume hinweg übertragen) und damit bewahren/konservieren lassen, bildet den Ausblick der Überlegungen. Damit widmet sich dieser Beitrag insgesamt Fragen, wie sie auch für das Navigationelle von Relevanz sind – insofern Orte, Wege, Pfade, Adressen, Adressierbarkeiten und Strukturen im Fokus stehen –, allerdings in einem Raum, in dem Navigation selten explizit wird: dem technologischen Mikroraum.

2. WAS IST SPEICHER?

Während die Informationsübertragung *Raumräume* miteinander verbindet, überbrückt deren Speicherung *Zeiträume*. Bei genauerer Überlegung zeigt sich, dass damit bereits jede Übertragung auch einen Speichervorgang darstellt, denn *während* Information übertragen wird, vergeht Zeit. Ein Ziel der Entwicklung von Übertragungstechnologien war und ist daher die Minimierung dieses unintendierten Zeitverlustes, dem jedoch physikalisch-bedingte Grenzen gesetzt sind.⁴ Technische Informationsspeicherung machte aus dieser Not eine Tugend, als sie aus der Übertragungstechnologie hervorging, denn »[e]ine Verzögerung von Informationen muss nicht zwingend stören, sondern kann genauso gut eine positive Eigenschaft sein, denn durch sie werden aus Übertragungssystemen dynamische Speicher.«⁵ Solche Verzögerungsspeicher, wie z. B. Quecksilber- oder Drahtspeicher bilden

2 Kittler: *Aufschreibesysteme*, S. 519.

3 Völz: *Speicher als Grundlage für alles*.

4 Keine Information kann schneller als mit Lichtgeschwindigkeit übertragen werden.

5 Maibaum: »Lumped Lines«, S. 147f.

die frühesten Binärspeicher für Computersysteme. Sie basieren auf den physikalischen Eigenschaften der Speichermedien (etwa der Verzögerung, mit der ein akustischer Impuls durch einen aufgewickelten Draht läuft). (Vgl. Abb. 1)

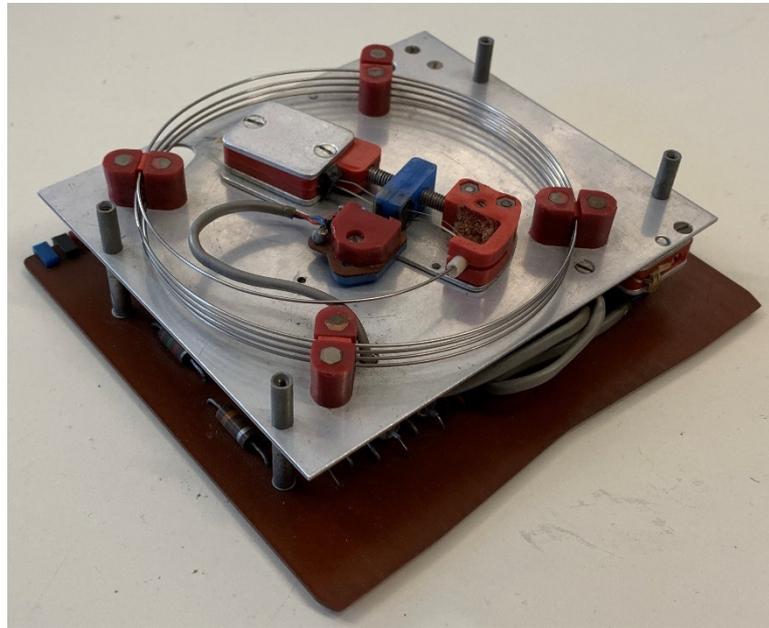


Abb. 1: Drahtspeicher (Medienarchäologischer Fundus, eigenes Foto).

Im 20. Jahrhundert entstehen in rascher Abfolge weitere Speichertechnologien, die sich nach drei Kategorien ordnen lassen: nach der *Speicherdauer*, nach der *Speichergeschwindigkeit* und dem *Energieverbrauch*, der für die Speicherung nötig ist. Je nachdem, welcher dieser Parameter der bestimmende oder begrenzende Faktor ist, werden die Speicher als Kurzzeit- oder Langzeitspeicher, innerhalb oder außerhalb von Medien eingesetzt, sind mit kleinem oder größerem Aufwand lesbar, beschreibbar oder löschar und können wahlfrei oder nur seriell adressiert werden.

Die *Adressierbarkeit* von Speicherinhalten ist ein entscheidendes Kriterium für ihren Einsatz in Computertechnologien. Speicher erhalten durch technische Ergänzungen (»Randlelektronik«⁶) neben der in ihnen gespeicherten Information eine Metainformation über den Speicherort, also die Adresse, unter welcher eine Information in ihnen abgelegt ist. Diese Adressierbarkeit ist, wie die Frage, ob diese Adressen linear vorgegeben sind oder vom Medium berechnet werden können, ein entscheidendes Kriterium für die »Mächtigkeit«⁷ eines Systems (wenn dieses Sys-

6 Völz: »Informations- und Speichertheorie«, S. 231-234.

7 Haigh: »Von-Neumann-Architektur«, S. 133f.

tem Berechnungen anstellen können soll). Die Adressierbarkeit gespeicherter Informationen sorgt für deren (im messtechnisch-topografischen Sinne⁸) *Lokalisierbarkeit*, gibt ihnen einen *Ort*, der sich bestimmen lässt.⁹ Die arbiträre Verbindung von Adresse und Information macht aus dem Vorgang der Speicherung einen *strukturellen* Prozess.

Durch die Zeitverzögerung gleicht das Abrufen einer adressierbaren Information aus dem Speicher dem Abschreiten eines Weges von der Senke zur Quelle des gespeicherten Datums. Solche Wege, die die Information dabei zurücklegt – technisch realisiert als Busse und Leitungen, diagrammatisch beschrieben als Datenpfade und -flussdiagramme –, wären auch topografisch darstellbar. Eine solche Wegbeschreibung bliebe allerdings notwendig unvollständig, denn die Errungenschaft des speicherprogrammierbaren Computers ist es, solche Topografien nicht bloß statisch, sondern dynamisch her- und zur Verfügung zu stellen. Dabei stellt sich die konkrete Ausgestaltung eines Pfades durch eine ihn aktivierende Operation her. Das Bild einer solchen Struktur wäre damit die *Architektur als technische Struktur*¹⁰ – diese sagt aber nichts über ihren konkreten Sinn aus, denn sie bildet bloß den Möglichkeitsraum¹¹ aller verfügbaren Pfade. Die konkrete Realisation entfaltet sich hingegen erst zur Laufzeit des Systems und kann zum Beispiel in der messtechnischen Erfassung der aktivierten Leitungen und Bauteile zu einem spezifischen Zeitpunkt Auskunft über die jeweilige Medienfunktion geben – »computer architecture [...] is both structure and process.«¹²

Auf diese Weise zeigt sich ein solcher Möglichkeitsraum dadurch, dass er eine konkrete Menge von Schaltungswegen realisierbar macht (und alle anderen ausschließt), ebenfalls als eine Form des Speichers, die Völz bei seiner eingangs zitierten Aufzählung noch unberücksichtigt lässt, die aber wohl einen Großteil der auch heute noch genutzten technischen Speicher umfassen würde. Dieser *Strukturspeicher* erweitert die Speichertheorie, die nach dem *knowing what* (Daten) und dem

8 Topographie verstanden als »Ortsbeschreibung, die Lehre vom begrifflichen und messtechnischen Erfassen, analogen und digitalen Modellieren und Darstellen des Georums, seiner Objekte und ihrer Bezüge (z. B. topographische Objekte, Reliefformen).« (<http://txt3.de/strukturspeicher1>, 09.06.2021)

9 Höltgen: »JUMPs durch exotische Zonen«, S. 120-125.

10 Der Medienarchäologe Wolfgang Ernst verwendet die Begriffe »Struktur« und »Architektur« synonym: »Architecture refers to physical building materialities and to structural conditions, the arché.« (Ernst: »Towards a Museology«, S. 49) Die terminologische wie medienarchäologische Gleichursprünglichkeit führt er auf Überlegungen Walter Seitters zurück: »Man muß architektonisch sehen, das heißt lesen. Ein Buch ist etwas anderes als ein Faden der Erzählung. Das lateinische Wort für Architektur heißt Struktur. Strukturalismus – besser: Strukturalistik heißt: daß man architektonisch sieht. In unserer Zeit muß das extra versucht werden.« (Seitter: »Das politische Wissen«, S. 60. zit. n. Ernst: »Medium Foucault«, S. 84.)

11 von Hilgers: »Zur Einleitung«, Ss. 9, 19, 23.

12 Ernst: »Towards a Museology«, S. 50.

knowing how (Algorithmen) ein Konzept des *knowing where* erforscht: Nämlich die konkreten Räume und Orte der Datenspeicher und wie diese mikro-logistisch organisiert sind. Die Strukturen, mit denen, in denen und durch die gespeichert wird, erhalten aus dieser Perspektive den Status eines *spatiologischen Aprioris* der Speicherung – und erweisen sich mithin als eine strukturelle, architektonische Bedingung der Navigation im technologischen Mikroraum, die überhaupt erst über die Effizienz technologischer (Speicher-)Systeme bestimmt.

3. DAS SPEICHERN DER SPEICHER: STRUKTURSPEICHER-BEGRIFFE

Strukturspeicher sind also konkrete Verschaltungswege für Informationen. Diese stellen als Möglichkeitsraum eine endliche Menge an Pfaden zur Verfügung, können darüber hinaus aber selbst entweder *fix/strukturiert* oder *flexibel/strukturierbar* sein. Beide Eigenschaften werden bereits in der Literatur zu Strukturspeichern beschrieben. Zuerst taucht der Terminus »Strukturspeicher« in der Lernmatrixtheorie Karl Steinbuchs um 1956 auf:

Bei den »Strukturspeichern« werden die elektrisch angebotenen Informationen in Strukturänderungen starrer Körper umgesetzt. Zu den Strukturspeichern gehören demnach alle Anordnungen, die von ferromagnetischem oder ferroelektrischem Verhalten Gebrauch machen, ebenso Speicher, welche die Schwärzung photographischer Schichten ausnutzen. Strukturspeicher haben den großen Vorzug, von dauernder Energiezufuhr unabhängig zu sein und hohe räumliche Speicherdichten zu ermöglichen.¹³

Steinbuch beschreibt hier Magnetkernspeicher, also Datenspeicher, bei denen die Speicherung durch eine Umstrukturierung magnetischer Eigenschaften erreicht wird: In Magnetspeichern *kippen* die magnetischen Zustände durch den Einfluss eines von außen angesetzten Magnetfeldes (Barkhauseneffekt¹⁴) in die eine oder die andere Richtung (Hysterese¹⁵) und lassen sich so als distinguierbare Speicherzustände definieren. Die auf diese Weise generierten Speicherungen sind oft sehr dauerhaft. Hier werden allerdings *Inhalte* als Strukturveränderungen gespeichert; die so erzeugten *Strukturen selbst* sind dabei irrelevant und können daher auch latent bleiben. Sobald allerdings die speichernden Systeme nicht (mehr) verfügbar und damit die Speicherinhalte nicht mehr (mit diesen) auslesbar sind, ändert sich die Situation und die latente Speicherstruktur gerät selbst in den Fokus der Preservation, um über die Strukturänderung des Speichersubstrates die Speicherinhalte

13 Steinbuch: *Automat und Mensch*, S. 106; zuvor bereits in: Steinbuch: *Elektrische Gedächtnisse*, S. 9.

14 Völz: »Informations- und Speichertheorie«, S. 237-240.

15 Ebd., S. 242-245.

wiederzugewinnen. Bei Ringkernspeichern (vgl. Abb. 2) kann dies zum Beispiel mittels Ferrofluid-Visualisierung geschehen.

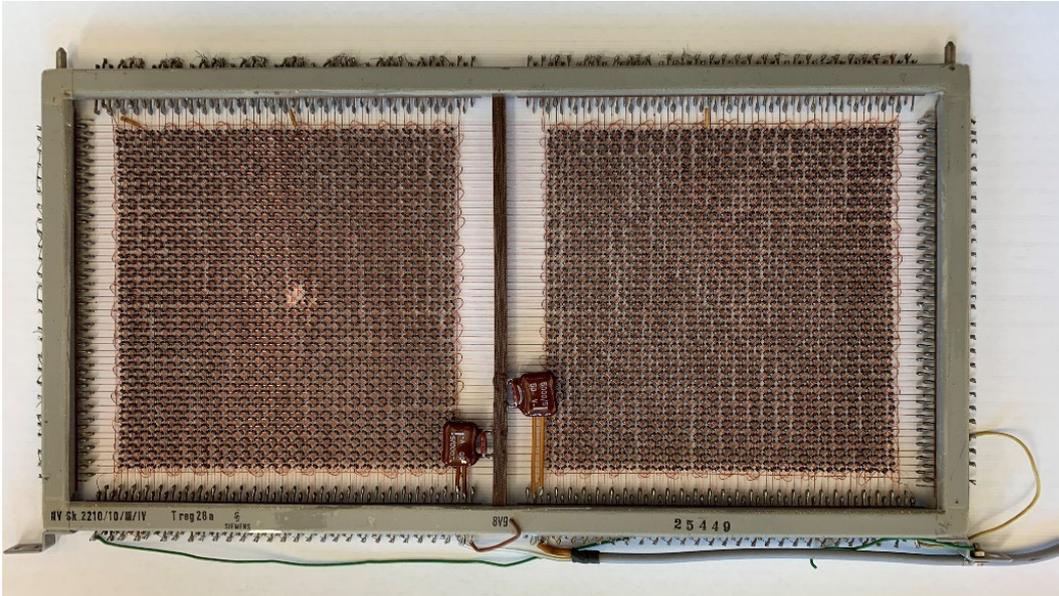


Abb. 2: Siemens S2002, 1000-Bit-Ferritkernspeicher (2 x 20 x 25 Bit), 2-mm-Kerne (ca. 1960). (eigenes Foto)

Steinbuchs Definition von Strukturspeicher, verstanden als (*reversible*) *Struktur zum Speichern*, greift die Fachliteratur in seiner Nachfolge auf¹⁶, um damit eine dedizierte Speichergattung zu kategorisieren. Winfried Hahn schreibt 1971 in seinem *Elektrotechnik-Praktikum für Informatiker*:

Bei Strukturspeichern letztlich wird von der Tatsache Gebrauch gemacht, daß bei manchen festen Körpern durch elektromagnetische Felder, Elektronenbestrahlung u. ä. die Struktur in umkehrbarer Weise veränderlich ist. Die bekannteste umkehrbare Änderung dieser Art ist die ferromagnetische Hysterese, die die Grundlage für Magnetband-, Magnettrommel- und Ferritkernspeicher bildet, die z. Z. wichtigsten Speicher datenverarbeitender Anlagen.¹⁷

Zwanzig Jahre später ist der Begriff in der Elektrotechnik und technischen Informatik schließlich lexikalisiert und der Strukturspeicher damit in den Kanon technischer Datenspeicher aufgenommen: »Zur Realisierung digitaler Speicher bzw. der Zu-

¹⁶ Wengleich ohne die von Steinbuch intendierten Anwendungen in einer *Lernmatrix mit bedingten Reflexen* (Steinbuch: *Automat und Mensch*, S. 172).

¹⁷ Hahn: *Elektronik-Praktikum*, S. 121f.

griffsmöglichkeiten zu den einzelnen Speicherzellen wurden erfolgreich mechanische (Strukturspeicher, Lochband, Bewegung von Medien) [...] u.a. Wirkprinzipien untersucht.«¹⁸

Mitte der 1970er-Jahre¹⁹ erfährt der Begriff innerhalb der Informatik jedoch eine Neubedeutung: Datenflussrechner²⁰ speichern Programme in Form von Datenflussdiagrammen in globalen Strukturspeichern.²¹ Hier sind es nun die virtuellen Datenstrukturen selbst, die gespeichert werden – allerdings in konventionellen RAM-Bausteinen, die in den Datenflussrechnern verbaut sind. Es handelt sich also um Software, die Struktur-Information beschreibt.

1993 findet eine weitere, ähnliche Neubesetzung des Begriffs statt, als der Informatiker Ingolf Grieger den Strukturspeicher als »ein[en] konzeptionell arbeitsplatzunabhängige[n] Speicherbereich für Strukturnetzwerke«²² definiert. In diesem können »graphische Daten, die in Einheiten genannt Strukturen, organisiert sind [...], miteinander hierarchisch in Beziehung stehen und bilden dann ein Strukturnetzwerk. Jede Struktur ist durch einen eindeutigen von der Anwendung spezifizierten Strukturnamen gekennzeichnet.«²³ Hier werden also Strukturbeschreibungen gespeichert, um grafische Daten später wieder zur Verfügung zu stellen (z.B. zur Ansicht zu bringen). Abermals wird hier mit Strukturspeicher eine in Software/Daten *kodierte Struktur* gemeint. Unter diesem Verständnis ist der Begriff bis heute innerhalb der grafischen Datenverarbeitung einschlägig geblieben.

Gemein ist allen dreien Strukturspeicher-Konzepten, dass entweder die Umstrukturierung von Substrat als Speicher genutzt wird oder strukturelle Beschreibungen konventionell gespeichert werden. Ist im ersten Fall die dabei entstehende materielle Struktur selbst irrelevant, so genügt im zweiten Fall das Speichern der funktionsäquivalenten *Strukturbeschreibung*, um diese später virtuell zu rekonstruieren. 2018 scheint Horst Völz beide Konzepte in seinem Begriff des Strukturspeichers zusammenzuführen:

Für technische *Strukturspeicher* gibt es hauptsächlich drei Varianten, die als PLD (programmable logic device, ähnlich dem FPGA = field-

18 Rhein/Freitag: Mikroelektronische Speicher, S. 2.

19 Ackerman: A Structure Memory.

20 Datenflussrechner stellen eine alternative Architektur gegenüber den heute zumeist immer noch gebräuchlichen Von-Neumann-Rechnern dar. Das Grundprinzip der Datenflussrechner basiert auf der parallelen Verarbeitung von Informationen durch gleichzeitiges Ausführen von Maschinenbefehlen in Threads. Datenflussrechner benötigen keinen Programmzähler (der die serielle Abarbeitung kontrolliert) und keinen zentralen Speicher für Programm und Daten. Anstelle dessen residieren aktuelle Daten als Nachrichten, die in Registern zwischengespeichert werden. Das Konzept wurde Mitte der 1970er-Jahre entwickelt; einige Computer sowie Programmiersprachen, die nach dem Datenflussrechnerprinzip verfahren, wurden entwickelt (vgl. Ungerer: Datenflußrechner, S. 11-15.)

21 Ebd., S. 72.

22 Grieger: Graphische Datenverarbeitung, S. 233.

23 Ebd., S. 166.

programmable gate array) bezeichnet werden. Sie leiten sich allgemein aus einer Reihenschaltung von AND- und OR-Matrizen ab. Dabei werden n Eingangs- und m Ausgangssignale benutzt. Bei der üblichen (Masken-)Programmierung werden Dioden eingefügt oder weggelassen [...]. Dadurch sind n verschiedene Funktionen programmierbar. Aus dem PLA [programmable logic array, SH] entsteht ein PROM, wenn die AND-Matrix als Address-Encoder festgelegt wird. Je nach der Kombination der n Eingänge wird ein programmiertes (gespeichertes) Signal der Wortbreite m ausgegeben. Für Redundanzfreiheit (Vollständigkeit) muss dabei $m = n$ gelten. Die dritte Variante ist ein PAL (programmable array logic). Bei ihr stehen die verfügbaren Ausgangssignale fest, sie werden mittels programmierbarer Eingangssignale ausgegeben.²⁴

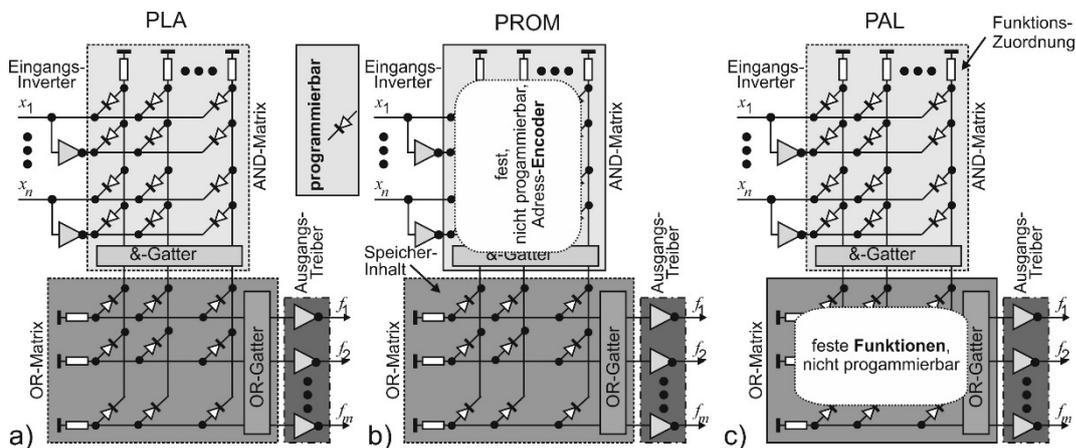


Abb. 3: PLDs: PLA, PROM, PAL.²⁵

Mit solchen Speichern sind sowohl Informationen (PROM) als auch das Speicherwissen (PLA, PAL) über Strukturierungsprozesse selbst speicherbar geworden. Sie verdoppeln damit die Grundeigenschaften der Rechenmaschine, die nun nicht mehr allein ihre Daten (Rechenwerte) und ihr Programm (Rechenplan), sondern auch ihre eigene Architektur (Rechenwerk) disponibel hält. Heinz von Foerster sieht die zentrale Eigenschaft adaptiver Rechensysteme darin, dass sie die Fähigkeit besitzen prinzipiell alle Berechnungen auf Basis eines Mechanismus durchführen zu können, ohne die Ergebnisse (*knowing that*) bereits vorliegen zu haben. In diesen Mechanismen ist das Prozesswissen (*knowing how*) selbst kondensiert und »in struktureller Weise gespeichert«.²⁶

24 Völz: »Informations- und Speichertheorie«, S. 235f.

25 Ebd., S. 236.

26 Von Foerster: »Gedächtnis ohne Aufzeichnung«, S. 134.

4. SPEICHERN ALS STRUKTURALISTISCHE TÄTIGKEIT

Der Strukturbegriff, wie er von den hier zitierten Technikern genutzt wird, beschreibt ein Gefüge aus miteinander verbundenen Elementen: Zu einer Struktur gehört »1. die Menge bzw. Klasse von Elementen, die als Strukturelemente fungieren, 2. das System syntaktischer Regeln, die die Ordnungseinheit [...] definierbar und erzeugbar macht, und 3. die Gesamtheit der über der Strukturbasis, diese repetierend, bildbaren Muster bzw. Pattern. Offensichtlich handelt es sich semiotisch um ein iconisches (Rahmen-)System.«²⁷ Wie bei der Adressierung von Informationen im Speicher bereits beschrieben (und wie sich im Begriff der *Adresse* schon andeutet), »kann man die Zeichenbegriffe [und damit die Struktur] auch topologisch (bzw. zeichentopologisch) erklären«²⁸: Voneinander getrennte Elemente werden zueinander in Beziehung gesetzt und diese Beziehung lässt sich topografisch darstellen. Entlang der Struktur (Graph, Operand, Leiterbahn, die zwei materielle oder immaterielle Objekte miteinander verbinden) bewegen sich die zugehörigen Informationen. Es findet also eine *Trennung von Speichersubstrat und Speicherinhalt* statt – eine Dichotomie, die an das *strukturalistische Paradigma der Trennung von Form und Inhalt* erinnert.

Deleuze sieht »[d]as Lokale oder die Stellung«²⁹ als wichtiges Erkennungskriterium für den Strukturalismus. Bei dieser Stellung »handelt [es] sich nicht um einen Platz in einer realen Ausdehnung, noch um Orte in imaginären Bereichen, sondern um Plätze und Orte in einem eigentlich strukturellen, das heißt, topologischen Raum.«³⁰ Dieser Raum wird konstituiert durch *Zeichen*: »In Wirklichkeit gibt es keine Struktur außerhalb dessen, was Sprache ist, und sei es auch eine esoterische oder sogar eine nicht verbale Sprache. [...] Die Dinge selbst haben nur insofern Struktur, als sie einen schweigenden Diskurs abhalten, welcher die Sprache der Zeichen ist.«³¹

Der Strukturbegriff des Strukturalismus geht aus der Sprachwissenschaft hervor – und die moderne Sprachwissenschaft wiederum aus dem Strukturalismus. Mit der Beschreibung der Beziehung von Wort (Signifikant) und Bedeutung (Signifikat) überführte Ferdinand de Saussure und in seiner Folge die strukturelle Linguistik, die Semiotik und alle anderen strukturalen und poststrukturalen Schulen das Denken in einen mentalen Bewegungsprozess – eine Mnemotechnik³² – und eine semiotische *Beziehungsarbeit*, die suchend die Verbindungen abschreitet, an welche Episteme gekoppelt sind oder wurden. Von dieser Methode sind auch die Formal-, Natur- und Technikwissenschaften bestimmt. Ob Mathematik (Mengen und deren

27 Bense »Struktur«, S. 104.

28 Bense »Raum, semiotischer«, S. 80.

29 Deleuze: *Woran erkennt man den Strukturalismus?*, S. 15.

30 Ebd.

31 Ebd., S. 8.

32 Wenz: »Linguistik/Semiotik«, S. 213.

Abbildungen), Physik (Raumzeitstruktur), Chemie (Strukturformeln), Logik (Junktoren), Elektrotechnik (Schaltungen), Informatik (Flussdiagramme, Strukturbäume) usw.: Sie alle besitzen Strukturkonzepte und -begriffe, die dem ursprünglichen Gedanken folgen, dass Sinn »notwendig und einzig aus der Stellung hervorgeht«³³.

Ebenso basiert das Konzept des Strukturspeichers, der seinen Sinn als Zeichensystem und (operatives) Diagramm entfaltet, auf – nun aber in Materie kondensierter – Kopplung: Abb. 3 zeigt schematisch, wie dieses Diagramm organisiert ist: Durch Dioden-Matrizen werden Wege für den Spannungsfluss freigegeben (1) oder gesperrt (0), mithin wird der mögliche Weg des Spannungsflusses materiell vorbestimmt und die Möglichkeit der Signal-Navigation dadurch limitiert. Der *Sinn* dieser Flusststeuerung liegt in der Herstellung von Schaltnetzen, bestehend aus AND- und NOT-Gattern, mit denen sich jede mögliche logische Struktur generieren lässt.³⁴ Der ursprüngliche Bezug zur Sprache, der in den technischen Beschreibungen von Schaltlogik oft unerwähnt bleibt³⁵, zeigt sich in ihrer Äquivalenz zur Aussagenlogik deutlich: Seit Aristoteles ist Logik ein Werkzeug zur Sprachanalyse. Über die Programmierung von FPGA-Strukturen mit Hilfe von Tools wie VHDL (Very High Speed Integrated Circuit Hardware Description Language) oder Verilog wird diese linguistische Basis als linearisierte Mnemotechnik wieder les- und schreibbar. Mit diesen Sprachen können logische Schaltnetze entworfen aber auch simuliert werden; Schaltnetze, die als – faktische oder hypothetische – Lagepläne auf Signalebene der operativen Wegfindung systematisch vorgeschaltet sind.

5. DIE GESTELLE DER MAKROSTRUKTUREN

Computer sind Apparate, die aus Theorie hervorgegangen sind – genau genommen aus formalsprachlichen Beschreibungen mathematischer Probleme: Als das Diagramm einer universellen Rechenmaschine bei Alan Turing oder als symbolischer

33 Ebd., S. 15.

34 Höltgen: »Logik«, S. 54.

35 Diese Tradition reicht bis zu Claude Shannons erstmaliger Formulierung der Schaltlogik als Schaltalgebra zurück. Ebenso Konrad Zuse, der im selben Jahr (1937) seinen Computer Z1 baut: »Für die Beschreibung der Schaltungen hat Zuse eine eigene graphische Darstellung erfunden und die logische Beschreibung hat er Bedingungskombinatorik genannt. Erst später hat er entdeckt, daß diese nur ein anderer Name für die Aussagenlogik war.« (Rojas u.a.: »Konrad Zuses Plankalkül«, S. 216)

»Plankalkül«³⁶ bei Konrad Zuse: Beide beschreiben ein *Gestell*³⁷, das aufgrund seiner arithmetischen, logischen und algorithmischen Fähigkeiten alles Berechenbare berechnen kann. Turing formulierte den Rechenvorgang als Anweisung (imperativ) an die Maschine, Zuse erklärte der Maschine (funktional-deklarativ), wie ein Problem zu lösen ist. Während die Turingmaschine eine Idee geblieben ist (die aufgrund ihrer Langsamkeit zu unpraktisch und aufgrund ihres unendlichen Speichers zu idealistisch *gedacht* war), leitete Zuse seinen Plankalkül aus der konkreten Implementierung seiner mechanischen und elektromechanischen Computer ab. Im Plankalkül finden sich deshalb strukturelle Beschreibungen von Computerfunktionen, die sich wiederum in Hardware (der *Logistischen Maschine*) sublimieren lassen sollten.

Die *strukturelle Beschreibung der Maschine* und die *Struktur der Maschine* wurden damit funktional äquivalent; und dennoch fehlt der sprachlichen Beschreibung gegenüber der physischen Maschine dasselbe wie dem Wort gegenüber dem ihn bezeichnenden Gegenstand: Die *Materialität*, die räumliche Konkretisierung, die bei operativen Medien stets eine *raumzeitliche* »Vergegenwärtigung«³⁸ ist. In beiden ist jedoch etwas gespeichert, das über ihr Speichervermögen selbst hinausgeht: Ihre Architektur, die jenen genannten Möglichkeitsraum der Informationsspeicherung, -übertragung und -verarbeitung abbildet, aus dem sich erst im operativen Zustand zu einem definierten Zeitpunkt eine der Möglichkeiten als Wirklichkeit realisiert. Diese Variabilität, die die inoperative Hardware nur andeuten kann, lässt sich in ihrer sprachlichen Beschreibung mental in Vollzug setzen. In Zuses Plankalkül hießen Variable in diesem Sinne »variable Strukturzeichen«³⁹, denn in Computern operiert »technische Semiotik«, wie Max Bense darlegt: Deren »mechanische Semantik wird an den Zustandsänderungen verifiziert, die der Text in einer wirklichen oder gedachten Maschine hervorruft, und auf diese Weise ebenfalls formalisierbar.«⁴⁰

Die sukzessive Verkleinerung von Rechnerstrukturen (Moore's Law) und die »Dissimulation«⁴¹ der Computer (Ubiquitous Computing) haben auch dazu geführt, dass diese Architekturen aus dem Blick geraten sind. Ein Blick in die Geschichte der Technologie zeigt aber schnell, dass sich gerade an der Struktur der Rechenapparate und Computer Episteme ablesen lassen, die eine ganz andere als

36 Als früheste höhere Programmiersprache verfolgt Zuses Plankalkül zunächst das Ziel die konkrete maschinensprachliche durch eine abstraktere symbolische Beschreibung einer logistischen Maschine zu ersetzen – eine Maschine, die er selbst nie gebaut hat. Erst 2000 haben Raúl Rojas und seine Kollegen einen Compiler für den Plankalkül geschrieben, der herkömmliche Digitalcomputer zu einer solchen logistischen Maschine macht (Rojas u.a.: »Konrad Zuses Plankalkül«).

37 Gestell als gedachte technische Struktur im Sinne Heideggers (Heidegger: »Die Frage nach der Technik«, S. 21f.)

38 Ernst: Signale aus der Vergangenheit, S. 169f.

39 Bauer/Wössner: »The »Plankalkül« of Konrad Zuse«, S. 679.

40 Baumgärtner: »Sprache und Automat«, zit. n. Bense: »Semiotik, technische«, S. 95.

41 Ernst: »Towards a Museology«, S. 56.

die ökonomische oder bürokratische Geschichte der Computer erzählen könnten. An zwei konkreten Beispielen sei dies exemplifiziert:

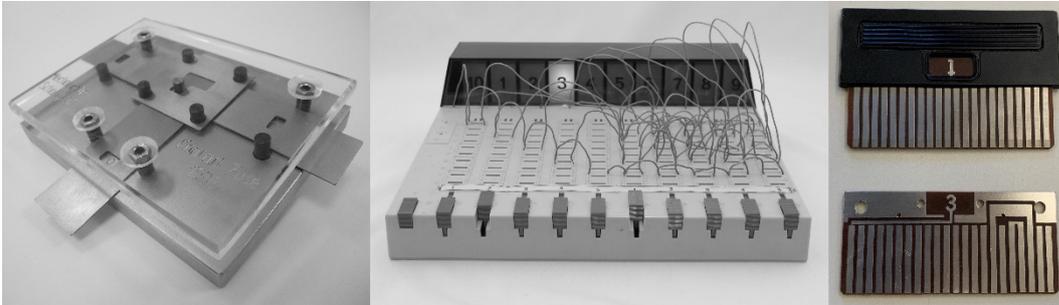


Abb. 4: Halbaddierer aus Zuses Z1 (links), Kosmos LOGIKUS (Mitte), Module für die Magnavox Odyssey (rechts). (links und Mitte: Fotos des Heinz-Nixdorf-MuseumsForums, Paderborn, rechts: eigenes Foto)

Erstes Beispiel: Konrad Zuses Computer Z1, gebaut zwischen 1935 und 1937, führt seine Rechenarchitektur als dreidimensionale mechanische Maschine vor – »the digital became architectural, in the sense of three-dimensional material structure.«⁴² Der Nachbau von 1995 (Abb. 4 links) lässt die raumgreifende Apparatur, die Zuse in seiner Berliner Wohnung aufgebaut hatte, kaum noch erahnen, kondensiert die Funktionen allerdings adäquat auf die kompakteren Maße eines Museumsexponats. Die Maschine ist funktional klar strukturiert. Die Mechanik der Z1 realisiert die Möglichkeit des Computers als bewegliche Struktur – dies zeigt sich nicht zuletzt daran, dass die funktionale Emulation von Z1-Komponenten als interaktive *Animation*⁴³ ihrer physikalischen Mechanik realisiert ist, deren Struktur äquivalent (s. o.) in linearem Code implementiert wurden. Durch die Auseinandersetzung mit der Struktur der Z1 hat Raúl Rojas offenbart, dass in der Architektur sogar mehr steckt, als Zuse seinerzeit intendiert hatte: Die Z1 ist im Prinzip turingvollständig; über einen *Hack* lassen sich ihre Rechenoperationen und Speichersysteme so verwenden, dass Schleifen und bedingte Sprünge ausgeführt werden können. Da die Z1 über keine symbolische Programmiersprache verfügt, konnte diese Möglichkeit allein aus ihrer physikalischen Struktur abgeleitet werden.

Zweites Beispiel: Videospiele-Plattformen benötigen nicht notwendigerweise eine CPU oder Software, damit man auf ihnen unterschiedliche Spiele spielen kann. Rudimentäre Frühformen solcher Geräte realisieren unterschiedliche Spiele allein mit Hilfe von Strukturveränderungen: Spielcomputer wie *Logikus* (1968, Abb. 4, Mitte) basieren auf einer variablen Verdrahtungslogik/Logikverdrahtung, bei der die

42 Ebd., S. 51

43 Von den Simulationen auf der Webseite des Zuse-Archivs lassen sich derzeit nur noch die WebGL-Anwendungen aufrufen: <http://txt3.de/strukturspeicher2>, 09.06.2021 (die Java- und Flash-Applikationen sind aus Alters- und/oder Sicherheitsgründen nicht aktivierbar).

Signalflüsse über Patchkabel auf den Oberseiten/-flächen des Gerätes gesteckt werden.⁴⁴ Mithilfe von Schiebeschaltern an der Vorderseite können die jeweils zehn Reihen mit An/Aus-Signalen versehen werden; über die Kabelstruktur, die zwischen den Reihen etabliert wird, realisiert sich ein konkreter Algorithmus. Ergänzbare Zeitgeber ermöglichen zusätzlich die *Programmierung* von getakteten seriellen und parallelen Operationen. Dadurch, dass sich über Drähte prinzipiell mehrere dieser Spielcomputer kaskadieren lassen, und mit diesen grundlegende logische Funktionen implementierbar sind, ließe sich ein turingvollständiges System aus ihnen konstruieren. Das Spiel (das hier »Wissen vermittel[n]«⁴⁵ heißt), wird dabei als (materialisierter) Spaghetticode auf der Oberfläche der Spielcomputer gespeichert und wäre damit *Softwire*. Anders sieht es bei der zur selben Zeit entwickelten *Odyssey*-Spielkonsole von Magnavox aus, die 1972 auf den Markt kam. Hier stehen die Spiele bereits im Möglichkeitsraum der Konsolenelektronik fest; sie müssen bloß noch durch geeignete *Hardwire* fertiggestellt werden (Abb. 4, Rechts). Dies erfolgt mittels kleiner Steckmodule, die nichts anderes als eine Leiterbahnstruktur (*Jumper*) enthalten. Die mit der Konsole ausgelieferten zwölf Spiele (auf je zwei Seiten von sechs Modulen) verschalten die Konsolenelektronik so, dass unterschiedliche Steuerungs- und Anzeigenereignisse stattfinden. Im Prinzip ließen sich durch bloße Permutation der 44 Modulkontakte (unter Berücksichtigung der elektrischen Definitionen der einzelnen Kontakte) weitere Spiele *entdecken*⁴⁶, die bereits latent in der Konsolenelektronik vorhanden sind.

Die hier vorgestellten Makrosysteme zeigen deutlich die strukturalen Zusammenhänge zwischen Architektur und Rechenfähigkeiten, zwischen Verdrahtung und Operativität oder – allgemeiner formuliert – zwischen Wegfindung und Wirkmächtigkeit der Systeme: Die materiellen Begrenzungen und Formatierungen der Möglichkeit des Navigierens auf Signalebene prägen je spezifische Räume des Spielens und Rechnens aus. In ihnen wird *spielerisch* sowohl die historische wie auch die epistemologische Entwicklung von Rechnern greifbar⁴⁷ und sichtbar. Rechenprozesse zeigen sich hier als mechanische Veränderungen – sowohl der rechnenden Strukturen als auch anhand der Bewegung von Signalen. Mit der Entwicklung der Mikroelektronik verschwinden diese Strukturen in elektronischen Schaltgattern auf Halbleiterbasis; aus den Schaltern, Wippen, Gestängen und Zahnradern werden Dioden und Transistoren, durch die Elektronen als Informationsentitäten gelenkt – navigiert – werden.

44 Sie stellen damit auch Modelle früher Patchkabel-programmierter Computer wie Colossus, ENIAC oder OPREMA dar.

45 Steinbuch: »Zum Geleit«.

46 Dies ist bereits geschehen: Das Spiel Tannhauser Gate wurde zum Beispiel 2019 als Spiel Nr. 13 von einem Game-Hacker entwickelt (<http://txt3.de/strukturspeicher3>, 09.06.2021)

47 Solche Systeme werden als unconventional computing auch für gegenwärtige und künftige Anwendungen erforscht (Adamatzky: Collision Based Computing).

6. MIKROSTRUKTUREN

Aber auch hier wird von Beginn an noch von Architekturen, »digital architectures«⁴⁸, gesprochen; diese meinen den Gesamtaufbau der Rechner, der einem bestimmten Grundprinzip folgt (Harvard, von Neumann, Datenflussrechner-Architektur usw.), oder die zentralen Recheneinheiten. Diese Mehrdeutigkeit des Architektur-Begriffs ist semiotisch begründbar:

Im Allgemeinen sind z.B. die Kommunikationskanäle oder Kommunikationsnetze eines städtischen Systems (Straßen, Energienetze, Informationsnetze) durch Zeichensysteme markiert und ausgezeichnet. [...] Als triadisches Objektsystem ist das urbane System als solches architektonisch stets konstituiert über dem (symbolischen) Repertoiresystem seiner metrischen bzw. parametrischen Maßverhältnisse, dem (indexikalischen) Richtungssystem seiner energetischen und kommunikativen Netzwerke und der Schließungswege und dem (iconischen) Rahmensystem des umbauten bzw. bebauten Raumes.⁴⁹

Von daher weisen die oft geäußerten Assoziationen von »close-up photographs of microchip circuits with bird's-eye views of a city«⁵⁰ auf die beiderseits vorhandenen Gemeinsamkeiten des strukturierten Informationsflusses hin. Raummetaphorik, das liegt schon im *Übertragungsbegriff* der Metapher (μεταφέρειν) begründet⁵¹, verdoppelt geradezu den strukturalen Sinn von Sprache als Informationstransportsystem: »Es lassen sich Raummetaphern in diesem Zusammenhang nachzeichnen, die digitale Räume als hochkomplexe Systeme vergleichbar mit Karten und Städten beschreiben.«⁵²

Um Chipstrukturen an die Oberfläche zu holen und wieder sichtbar zu machen, ist allerdings medienarchäologische Stratifikation⁵³ gefragt: In einem aufwändigen Prozess, bei dem einem in Kunststoff verpackten Chip mit chemischen und physikalischen »Grabschaufeln«⁵⁴ auf den Leib gerückt wird, wird dessen Struktur

48 Ernst: »Toward a Museology«, S. 49.

49 Bense: »Architektursemiotik«, S. 17.

50 Ernst: »Toward a Museology«, S. 70. Z.B. hier: »Computer Chips Look Like Intricate Cities«, <http://txt3.de/strukturspeicher4>, 09.06.2021.

51 Die Technikwissenschaften, insbesondere die Informatik, bedient sich systematisch der Metaphorik zur Umschreibung und ›Verschlagwortung‹ komplexer Prozesse (Busch: »Metaphern«). Die Reismetapher der Navigation findet sich diesbezüglich zum Beispiel in Erörterungen zu Theorien von Netzwerken (die abermals metaphorisch beschrieben sind) Anwendung (Büchel: »Praktische Informatik«, S. 239).

52 Wenz: »Linguistik/Semiotik«, S. 219.

53 Lang: »Archäologie«, S. 30.

54 Jan Engelmann spricht von der »Grabschaufel der Medienarchäologie« (Engelmann: »Aktenzeichen Foucault«, S. 225), die hier einmal im facharchäologischen Sinne zu verstehen wäre.

endlich wieder sichtbar. Deren Layer müssen mit weiteren chemischen und fotografischen Operationen separiert werden, um schließlich die »Archiv-Tektonik«⁵⁵ des Chips erkennbar werden zu lassen, die zeigt, wie der Rechner in seinem Kern aufgebaut ist. Am Beispiel der CPU Intel 4004 kann dies gezeigt werden (vgl. Abb. 5 – wenngleich die Unmöglichkeit, in die hier gedruckten Bilder hineinzuzoomen, schnell vor Augen führt, dass selbst dieser erste Mikroprozessor, der auf (nur!) quadratmetergroßem Papier entworfen wurde⁵⁶, schon enorme Komplexität auf kleinstem Raum verdichtet).

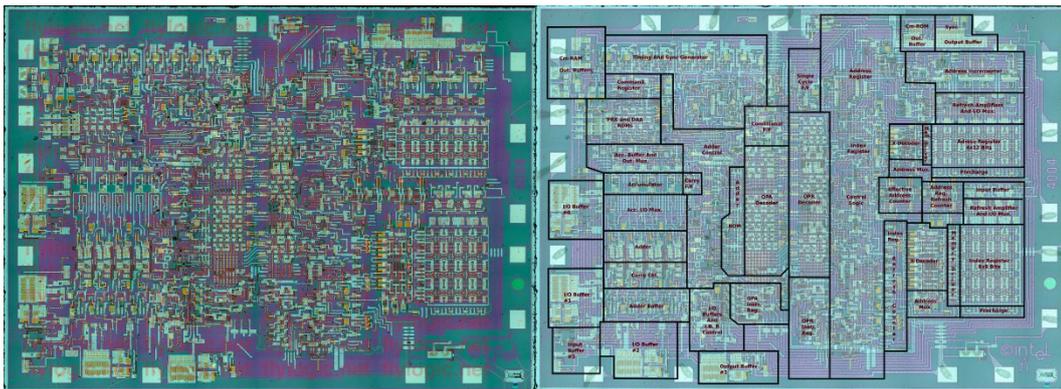


Abb. 5: Fotografie des Intel 4004⁵⁷ (links), Überlagertes Funktionsschema des Intel 4004⁵⁸ (rechts).

Eine CPU ist ein *semiotisches Design-Objekt* im Sinne Benses: Sie ist nicht bloß ein passiver Wegweiser, der den Signalweg vorgibt, sondern »ein System, das aus einem Träger und aktuellen Zeichen besteht.«⁵⁹ Die aktuellen Zeichen in der Intel-4004-CPU befinden sich jedoch *gefroren* (hartverdrahtet) im *Mikroprogramm*, zu dem ein *Befehlsregister* und *-dekoder* gehört (vgl. Abb. 6). Daneben enthält die CPU Speicher in Form von *Registern* (neben dem Programmzähler sowie drei 12-Bit-*Stacks* für Rücksprungadressen, einen *Akkumulator*, einen *Zwischenspeicher* und 16 4-Bit-Register die als *Indexregister* gepaart werden können). Die *Arithmetisch-Logische Einheit* (ALU) bildet das rechnende Herzstück des Prozessors: Alle Komponenten sind über einen 4-Bit-Datenbus miteinander verbunden und werden über einen 4-Bit-Steuerbus in ihrer Zusammenarbeit koordiniert. Die Befehle der 4004 sind allerdings 8 Bit *breit* und sie kann sogar mit 12-Bit-Adressen (vier Kilobyte) speichern; hierzu müssen die CPU-internen Befehls- und Adressmodi mit Multiplexern erweitert werden.

55 Ernst: »Toward a Museology«, S. 75.

56 <http://txt3.de/strukturspeicher5>, 09.06.2021

57 <http://txt3.de/strukturspeicher6>, 09.06.2021

58 <http://txt3.de/strukturspeicher6>, 09.06.2021

59 Bense: »Design-Objekt, semiotisches«, S. 24.

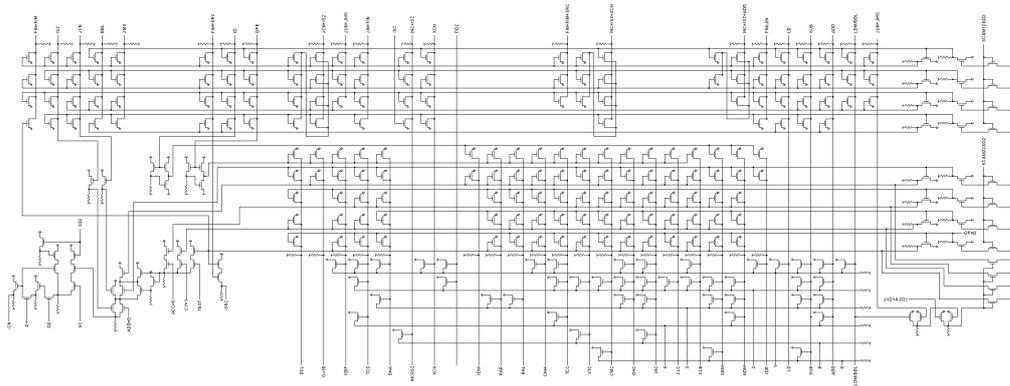


Abb. 6: Opcode-Matrix des 4004 als rekonstruiertes Schaltdiagramm.⁶⁰

Der Aufbau der 4004 ist, gerade wegen ihrer Bedeutung für die Mikrocomputer-Geschichte, weithin bekannt; es existieren zahlreiche Aussagen der Entwickler⁶¹, technische Entwicklungsunterlagen⁶², Datenblätter⁶³ und Fotografien des geschlossenen⁶⁴ und geöffneten⁶⁵ Chips. Die Strukturen der 4004 sind nahezu perfekt archiviert und bewahrt. Sie könnte jederzeit neu aufgelegt werden, sofern sich noch ein Hersteller findet, der die 2250 Transistoren und anderen Bauteile in PMOS-10-Mikrometer-Technologie fertigen kann. Seit der Entwicklung der LSI- und VLSI-Technologie sind jedoch unüberschaubar viele Bausteine erfunden und gefertigt worden, denen nicht dieselbe archivarische Aufmerksamkeit wie Mikroprozessoren im Allgemeinen⁶⁶ und Intels 4004⁶⁷ im Besonderen zuteil wurde. Für all diese müsste, sofern sich noch Exemplare von ihnen finden lassen, jener oben dargestellte stratigrafische Prozess nachvollzogen werden, um deren Strukturen sichtbar und erfahrbar zu machen.

7. SCHLUSS: SPEICHER-MEMORIALS

An dieser Stelle kehren wir zum Strukturspeicher zurück. Völz definierte ihn als einen Speicher, in welchem Strukturen angelegt (gespeichert) werden können. Diese Strukturen können alle Arten von digitalelektronischen Bausteinen sein. Die von Völz vorgestellten Strukturspeicher lassen sich als eine Form von *Struktur-RAM* sehen, in denen Strukturen angelegt und wieder gelöscht werden können. Mit Hilfe von FPGAs lassen sich beispielsweise beliebige logische Schaltnetze generieren und

60 <http://txt3.de/strukturspeicher7>, 09.06.2021.

61 <http://txt3.de/strukturspeicher8>, 09.06.2021.

62 <http://txt3.de/strukturspeicher9>, 09.06.2021.

63 <http://txt3.de/strukturspeicher10>, 09.06.2021.

64 <http://txt3.de/strukturspeicher11>, 09.06.2021.

65 <http://txt3.de/strukturspeicher12>, 09.06.2021.

66 <http://txt3.de/strukturspeicher13>, 09.06.2021.

67 <http://txt3.de/strukturspeicher14>, 09.06.2021.

replizieren – auch jene der Intel-4004-CPU⁶⁸. Hierzu bedarf es allerdings der Vorlagen, die in eine Beschreibungssprache übersetzt werden müssen (wobei nicht selten unbekannte Features und Design Flaws entdeckt werden⁶⁹), um daraus eine funktionsäquivalente FPGA-CPU zu generieren. Diese ist allerdings selten homotop zu ihrer Vorlage. Demgegenüber stellen fixierte Strukturen (ROM- und RAM-Bausteine, TTL-Bausteine, CPUs, ...) *Struktur-ROMs* dar – nicht veränderbare Strukturspeicher, in denen die Architektur als mikroelektronisches Diagramm *eingefroren* ist, durch das es in ähnlicher Weise zu navigieren gilt wie durch makrologische Stadträume. Gerade vor einem historiografischen und archäologischen Hintergrund erscheint es sinnvoll einen solchen fixierten Aufbau eines Bausteins, seine Mikroarchitektur, als *Monument* zu betrachten, das in seiner Architektur (Bauweise, Bauteile, ...) gleichermaßen auf seine Historiografie wie auf seine Funktion als Möglichkeitsraum verweist. Aus ihr ließe sich mit Hilfe einer »Archäologie [verstanden als] eine Raum- wie auch eine Zeitwissenschaft, in der räumlich und zeitlich die ›Dinge in der Welt‹ verknüpft werden«⁷⁰ Technikgeschichte ablesen, die alle Dimensionen eines solchen medientechnischen Objektes berücksichtigt und nicht allein auf mit bloßem Auge sichtbare Monumente beschränkt bliebe.

Es bedürfte nun allerdings zusätzlich ihrer Operativierbarkeit, um dieses Verständnis von »structure as process«⁷¹ zu erlangen. Diese kann auf unterschiedliche Weise erfolgen: Sofern der Baustein vorhanden und funktionstüchtig ist, lässt er sich in ein operatives Setting (re-)implementieren: Für Intels 4004 existieren einige Implementierungsvorschläge, die vom Nachbau des Busicom-Tischrechners⁷² bis hin zur Einbettung der CPU in ein Arduino-Shield⁷³ reichen. Dies erfordert jedoch die Erhaltung des Gehäuses, womit der Chip weiterhin unsichtbar bliebe. Daneben existieren Simulatoren, die die Signalflüsse auf der Chipoberfläche als Computergrafik animieren⁷⁴, und Emulatoren, die entweder die CPU in einem *historischen Setting*⁷⁵ oder in einer virtuellen Umgebung⁷⁶ emulieren. Diese Option besitzt alle Vor- und Nachteile einer Software-Nachbildung von Hardwarekomponenten: Es können nur solche Eigenheiten der Originalhardware emuliert werden, die den Programmierer:innen auch bekannt sind. Die Emulationstiefe und -genauigkeit ist dabei stets begrenzt und verfährt meistens nach dem Prinzip der Ausgabeadäquanz

68 <http://txt3.de/strukturspeicher15>, 09.06.2021.

69 Für den Intel 4004 sind im Verlaufe seiner Übersetzung in Verilog einige Design Flaws entdeckt worden: <http://txt3.de/strukturspeicher16>, 09.06.2021.

70 Lang: »Archäologie«, S. 42.

71 Ernst: »Toward a Museology«, S. 55.

72 <http://txt3.de/strukturspeicher17>, 09.06.2021.

73 <http://txt3.de/strukturspeicher18>, 09.06.2021.

74 Für drei Mikroprozessoren existieren bereits solche Simulationen unter <http://txt3.de/strukturspeicher13>, 09.06.2021.

75 <http://txt3.de/strukturspeicher19>, 09.06.2021.

76 <http://txt3.de/strukturspeicher20>, 09.06.2021.

zwischen Original- und emuliertem System. Die Emulation erlaubt allerdings auch Veränderungen, Erweiterungen und Ergänzungen, die das Originalsystem nicht zulassen würde.⁷⁷ So kann beispielsweise beliebig viel virtueller RAM-Speicher ergänzt oder ein wesentlich höherer Takt, als im Originalsystem möglich wäre, hinzu programmiert werden. Damit gerät die Emulation zu einem epistemischen Objekt, das es erlaubt, computerarchäologisch mit den Möglichkeiten und Unmöglichkeiten der Technikgeschichte zu spielen.

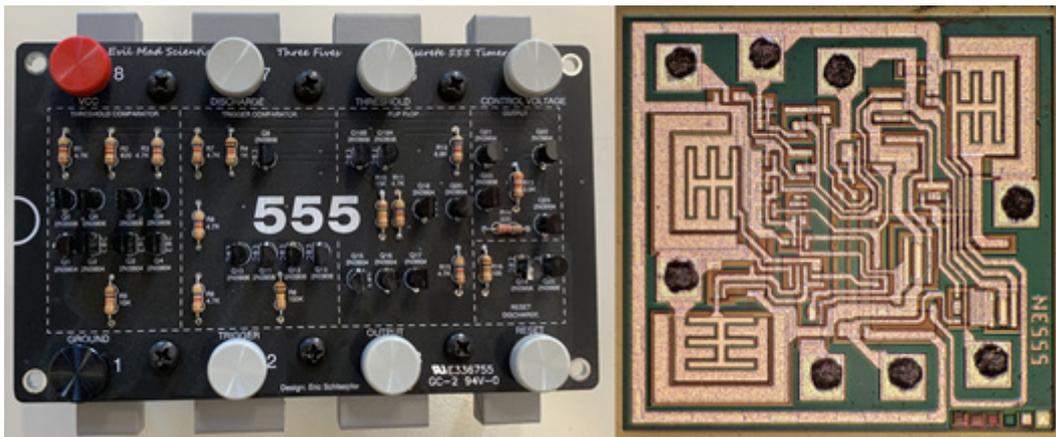


Abb. 7: Diskreter Nachbau des NE555 (links, eigenes Foto), Fotografie eines NE555⁷⁸ (rechts).

Softwareemulationen ermöglichen so zwar Operativerhaltung durch Aufhebung der Hardware in die Virtualität von Software, sie vermögen jedoch nicht die manifeste Struktur, die in ihnen gespeichert ist, zu bewahren. Dreidimensionale Chiplayouts werden durch sie in zweidimensionale Programmcodes transkribiert. Um diesem Verlust an Struktur zu begegnen, bieten sich andere Möglichkeiten an: Bausteine können als funktionale Schaltgatter real oder virtuell nachgebaut werden. Für CPUs, wie die MOS 6502⁷⁹ oder den Timerchip NE555 (Abb. 7), existieren Nachbauten, die die Originalstruktur mit diskreten Bauteilen (Transistoren) implementieren. Eine Möglichkeit Digitaltechnik regelrecht räumlich nachzubauen, bieten virtuellen Welten wie das Spiel »Minecraft«. Hierin kann mit Hilfe spieleigener Signalverarbeitungstechnologien (Redstone, Kolben oder Trichter) eine (virtuell) begehbbare Rechnerarchitektur konstruiert werden, mit denen die Strukturen der Vorlage nachgeahmt werden kann (aber nicht muss) (Abb. 8).

77 Höltgen: OPEN HISTORY, S. 195-201.

78 <http://txt3.de/strukturspeicher21>, 09.06.2021.

79 <http://txt3.de/strukturspeicher22>, 09.06.2021.



Abb. 8: Speicherwerk der Intel-4004-CPU in Minecraft⁸⁰ (r. u.: Handkompass zur Navigation).

Die damit erstellten Konstruktionen wären sogar lauffähig. Dass sie in Operation allerdings nur einen Bruchteil der Geschwindigkeit des Originals an den Tag legen und (verglichen mit der Größe des Avatars) gigantische Ausmaße aufweisen, kann einerseits als Hinweis auf den experimentell-analytischen (*toy computing*⁸¹) Zugang zur implementierten Technologie verstanden werden. Auch eine »Minecraft«-Welt ist bloß eine *archaeological map*, ein Datenraum, »der nicht physikalisch gebunden ist, sondern erst im Bezug auf ein Erkenntnisinteresse der Archäologen geschaffen wird.«⁸² Andererseits rufen die gigantischen virtuellen Rechnerbauten in »Minecraft« aber auch eine »computer-aided historical imagination«⁸³ der frühen Computer und Mainframes auf – Rechner, die von ihren Nutzer:innen tatsächlich *begebar* waren (wie etwa Grace Hoppers Debugging im Harvard Mark II⁸⁴ oder die Arbeiten der DDR-Ingenieure in der OPREMA⁸⁵ dokumentieren). Das Durchwandern solch virtueller, operativer und interaktiver Rechnerstrukturen verwirklicht damit zugleich eine *Mnemotechnik* wie es eine Archäologie ermöglicht (im ursprünglichen Sinn des Wortes als *Redaktion*), denn alles, was gesehen und beschriftet werden kann, kann über den »Minecraft«-Avatar auch verändert werden.

Eine so erstellte *Computerwelt* lässt sich wiederum abspeichern und auf einem anderen Computer in »Minecraft« laden und nutzen.⁸⁶ Die Operativität ist eine emulierte, deren Ausgaben nicht mehr symbolisch, sondern ikonisch in Form eines

80 Screenshot des Entwurfs: <http://txt3.de/strukturspeicher23>, 09.06.2021

81 Höltgen u.a.: »Toy Computing«.

82 Lang: »Archäologie«, S. 34.

83 Ernst: »Toward a Museology«, S. 68.

84 <http://txt3.de/strukturspeicher24>, 09.06.2021.

85 Winkler: »Oprema«, S. 9.

86 Die Intel-4004-Speicherarchitektur aus Abb. 8 und Abb. 9 kann unter diesem Link geladen werden: <http://txt3.de/strukturspeicher25>, 09.06.2021.

operativen Diagramms erfolgt. Letztlich stellt jede Übertragung eines Strukturspeichers auf ein anderes System oder Substrat eine *strukturalistische Tätigkeit* dar, die zeigt, wie tief sich der Strukturalismus in unser technisches Denken eingeschrieben hat – so tief, dass die Sprache regelrecht in Materie sublimiert ist. Die »Computer-aided architectural reconstruction«⁸⁷ mit »Minecraft« ist damit zuletzt spielerisches *Computer-aided Design*. Der Computer wird dabei zum Archiv und Museum seiner selbst, wie Wolfgang Ernst und Friedrich Kittler erinnern: »archaeology proposes a museological scenario in this specific sense of computer architecture [...] not simply [...] computer-augmented museum space and interactive virtual collections. In a shift away from simple interface metaphors, virtual reality can be applied in a different way; it allows us ›to enter [and to navigate, SH] the architecture of digital media«⁸⁸.⁸⁹



Abb. 9: Speicherwerk der Intel-4004-CPU in Minecraft – von innen (Bildmitte: Avatar des virtuellen Archäologen).

87 Ernst: »Toward a Museology«, S. 63.

88 Kittler: »Museums on the Digital Frontier«, S. 77.

89 Ernst: »Toward a Museology«, S. 56.

LITERATURVERZEICHNIS

- Ackerman, William B.: A Structure Memory For Data Flow Computers. (Master Thesis) Laboratory for Computer Science, Massachusetts Institute of Technology, 1977. Auch unter <https://dl.acm.org/doi/10.5555/889796>.
- Adamatzky, Andrew (Hrsg.): Collision-Based Computing, London 2002.
- Bauer, F. L./Wössner, H.: »The »Plankalkül« of Konrad Zuse: A Forerunner of Today's Programming Languages«, in: Communications of the ACM, Vol. 17, No. 7 (July 1972), S. 678-685.
- Baumgärtner, K.: Sprache und Automat. Information und Kommunikation. Berichte der 23. internationalen Hochschulwoche, Alpbach 1967.
- Bense, Max: »Design-Objekt, semiotisches«, in: Ders./Walther, Elisabeth (Hrsg.): Wörterbuch der Semiotik, Köln 1973, S. 24.
- Bense, Max: »Struktur«, in: Ders./Walther, Elisabeth (Hrsg.): Wörterbuch der Semiotik, Köln 1973, S. 104.
- Bense, Max: »Raum, semiotischer«, in: Ders./Walther, Elisabeth (Hrsg.): Wörterbuch der Semiotik, Köln 1973, S. 80.
- Bense, Max: »Semiotik, technische«, in: Ders./Walther, Elisabeth (Hrsg.): Wörterbuch der Semiotik, Köln 1973, S. 95.
- Bense, Max: »Architektursemiotik«, in: Ders./Walther, Elisabeth (Hrsg.): Wörterbuch der Semiotik, Köln 1973, S. 16f.
- Büchel, Gregor: Praktische Informatik – Eine Einführung, Wiesbaden 2012.
- Busch, Carsten: Metaphern in der Informatik. Modellbildung – Formalisierung – Anwendung. Wiesbaden 1997.
- Engelmann, Jan: »AktENZEICHEN Foucault«, in: Ders. (Hrsg.): Michel Foucault – Bot-schaften der Macht. Der Foucault-Reader. Diskus und Medien, Stuttgart 1999, S. 215-226.
- Ernst, Wolfgang: M.edium F.oucault. Weimarer Vorlesungen über Archive, Archäologie, Monumente und Medien. Weimar 2000.
- Ernst, Wolfgang: Signale aus der Vergangenheit. Eine kleine Geschichtskritik, München 2013.
- Ernst, Wolfgang: »Towards a Museology of Algorithmic Architectures«, in: Goodhouse, Andrew (Hrsg.): When is the Digital in Architecture?, Berlin 2017, S. 45-78.
- Grieger, Ingolf: Graphische Datenverarbeitung. Mit einer Einführung in PHIGS und PHIGS-PLUS. Zweite, vollständig neubearbeitete und erweiterte Auflage mit 120 Abbildungen, Berlin u.a. 1992.
- Hahn, Winfried: Elektronik-Praktikum für Informatiker, Berlin u.a. 1971.

- Haigh, Thomas: »Von-Neumann-Architektur, Speicherprogrammierung und modernes Code-Paradigma – drei Leitbilder früher Rechenanlagen«, in: Zeitschrift für Medienwissenschaft, Heft 12, 2015, S. 127-139. <https://doi.org/10.25969/mediarep/1362>, 09.06.2021.
- Heidegger, Martin: »Die Frage nach der Technik« [1953], in: Ders.: Vorträge und Aufsätze, Frankfurt a.M. 2000, S. 9-40.
- Höltgen, Stefan: »JUMPs durch exotische Zonen. Portale, Hyperräume und Teleportationen in Computern und Computerspielen«, in: Hensel, Thomas/Neitzel, Britta/Nohr, Rolf F. (Hrsg.): »The Cake is a Lie!« Polyperspektivische Betrachtungen des Computerspiels am Beispiel von »Portal«, Münster 2015, S. 107-134.
- Höltgen, Stefan: Logik für Medienwissenschaftler. In: Ders. (Hrsg.): Medientechnisches Wissen: Logik, Informations- und Speichertheorie, Bd. I. Berlin/Boston 2018, S. 14-149.
- Höltgen, Stefan: OPEN HISTORY. Archäologie der frühen Mikrocomputer und ihrer Programmierung. (Dissertation) 2020, <http://txt3.de/open-history>, 09.06.2021.
- Höltgen, Stefan/Pleikies, S./Fecker, T./Mormsbecher, N./Divani, S.: »A Case of Toy Computing. Implementing Digital Logics with Minecraft«, in: Adamatzky, Andrew (Hrsg.): Alternative Computing. (voraussichtlich) Heidelberg u.a. 2021, S. 539-580.
- Kittler, Friedrich: Aufschreibesysteme 1800-1900, 3. überarb. Aufl., München 1995.
- Kittler, Friedrich: »Museums on the Digital Frontier«, in: Keenan, Thomas (Hrsg.): The End(s) of the Museum. Barcelona 1996, S. 67-80.
- Lang, Franziska: »Archäologie«, in: Günzel, Stephan (Hrsg.): Raumwissenschaften. Frankfurt a.M. 2009, S. 30-45.
- Maibaum, Johannes: »Lumped Lines und Bucket Brigades – Verzögerungsleitungen als dynamische Speicher«, in: Ders./Ernst, Wolfgang (Hrsg.): Speicher. Theorie, Technologie, Archäologie. Ausgewählte Schriften von Horst Völz, Reihe: Computerarchäologie Mainframe, Bd. I, Bochum 2021, S. 147-164.
- Rhein, D. & Freitag, H.: Mikroelektronische Speicher. Speicherzellen, Schaltkreise, Systeme, Berlin 1992.
- Rojas, Raúl/Göktekin, Cüneyt/Friedland, Gerald/Krüger, Mike/Scharf, Ludmila/Kuniß, Denis/Langmack, Olaf: »Konrad Zuses Plankalkül – Seine Genese und eine moderne Implementierung«, in: Hellige, Hans Dieter (Hrsg.): Geschichten der Informatik. Visionen, Paradigmen, Leitmotive. Berlin/Heidelberg 2004, S. 141-170.
- Seitter, Walter: Das politische Wissen im Niebelungenlied, Berlin 1987.
- Steinbuch, K.: Automat und Mensch. Über menschliche und maschinelle Intelligenz, Berlin u.a. 1961.

- Steinbuch, Karl: Elektrische Gedächtnisse für Ziffern, in: Sonderdruck aus ETZ-A, Heft 21, 1. November 1956.
- Steinbuch, Karl: »Zum Geleit«, in: Lohberg, Rolf: Spielcomputer LOGIKUS, Stuttgart 1969, S. 4.
- Ungerer, Theo: Datenflußrechner, Stuttgart 1993.
- Völz, Horst: »Informations- und Speichertheorie«, in: Höltgen, Stefan (Hrsg.): Medientechnisches Wissen: Logik, Informations- und Speichertheorie, Bd. I. Boston/Berlin 2018, S. 150-282.
- Völz, Horst: Speicher als Grundlage für alles, Herzogenrath 2019.
- von Foerster, Heinz: »Gedächtnis ohne Aufzeichnung« [1965], in: Ders.: Sicht und Einsicht. Versuche zu einer operativen Erkenntnistheorie, Wiesbaden 1985, S. 133-172.
- Von Hilgers, Philipp: »Zur Einleitung. Eine Epoche der Markovketten«, in: Ders./Velminski, W. (Hrsg.): Berechenbare Künste. Andrej A. Markov. Mathematik, Poesie, Moderne, Zürich/Bern 2007, S. 9-27.
- Wenz, Karin: »Linguistik/Semiotik«, in: Günzel, Stephan (Hrsg.): Raumwissenschaften, Frankfurt a.M. 2009, S. 208-224.
- Winkler, Jürgen F. H.: Oprema – The Relay Computer of Carl Zeiss Jena, 2019, <https://arxiv.org/abs/1908.09549>, 09.06.2021.