

Humboldt-Universität zu Berlin
Kultur-, Sozial-, und Bildungswissenschaftliche Fakultät
Institut für Musikwissenschaft und Medienwissenschaft
Masterstudiengang Medienwissenschaft
Sommersemester 2015
Betreuer: Dr. Stefan Höltgen

Projektbericht

Retrocomputing als praktische Medienarchäologie

**Über die Entwicklung eines Flash-Speicher-Moduls
für die Videospielekonsole „Interton VC 4000“**

Johannes Maibaum

28. September 2015

Matrikelnummer: 533368

4. Fachsemester

Paul-Grasse-Str. 12, 10409 Berlin

jmaibaum@gmail.com

Inhaltsverzeichnis

1	Exposition des Problemfeldes	1
2	Die Aufgabenstellung	6
3	Der Lösungsweg	7
3.1	Analyse des Hardwareaufbaus	7
3.2	Erste Hardwaretests mit einem SRAM	12
3.3	Der dritte Zustand binärer Elektronik	14
3.4	Auf der Jagd nach Hardware-Fehlern	19
3.5	Zusätzliche Logikgatter	22
3.6	(SD-)Kartenspiele	23
3.7	Lesen und Schreiben	24
3.8	Das letzte fehlerhafte Spiel	26
4	Das Ergebnis	30
4.1	Arduino-Firmwarecode	34
5	Retrocomputing als medienarchäologische Methode	38
5.1	Medientheoretische Erkenntnisse durch aktives Retrocomputing	41
5.2	Medienarchäologie versus Mediengeschichte?	43
5.3	Senden und Empfangen	46
5.4	Fazit	47
	Abbildungsverzeichnis	48
	Literatur	49

1 Exposition des Problemfeldes

Die Aufgabenstellung des durchgeführten Projekts entstand während der Lehrveranstaltungsreihe „Signetics 2650 Programmier-Workshop“ bei Dr. Stefan Höltgen, die erstmals im Wintersemester 2013/14 und in den danach folgenden zwei Semestern angeboten wurde. Hier hatten Studierende der Bachelor- und Masterstudiengänge Medienwissenschaft die Möglichkeit, eine medienwissenschaftlich orientierte Einführung in die Assemblerprogrammierung zu erhalten.¹

Gegenstand des Programmierworkshops war der im Jahr 1975 vorgestellte 8-Bit-Mikroprozessor 2650.² Dieser wurde von der 1961 gegründeten Firma *Signetics* hergestellt, einem der ersten Hersteller kommerzieller integrierter Schaltkreise (*integrated circuits*, ICs).³ Noch im Jahr 1975 wurde *Signetics* vom US-Ableger des niederländischen Konzerns *Philips* gekauft, behielt aber den Namen *Signetics* bis 1993 bei, bevor sie in *Philips Semiconductors* umbenannt wurde. 2006 trennte sich *Philips* von seiner Halbleitersparte, die seitdem als Firmenkonsortium unter dem Namen *NXP Semiconductors* weiterhin ICs produziert und verkauft.⁴

Nachdem im ersten Teil des Kurses zunächst das von *Signetics* vornehmlich zu Schulungszwecken an ihrem Mikroprozessor entwickelte Lehrsystem *Instructor 50* programmiert wurde, wechselten wir ab dem zweiten Semester zur Videospielekonsole *VC 4000* der deutschen Firma *Interton Electronic*.⁵ Diese enthält neben dem 2650 als Hauptprozessor das *Programma-*

1 Vgl. hierzu Stefan Höltgen. „Die NOPs und HALTs digitaler Medien. Programmierlehre maschinennaher Sprachen für Medienwissenschaftler“. In: *Grundlagenstudien aus Kybernetik und Geisteswissenschaften. Internationale Zeitschrift für Modellierung und Mathematisierung in den Humanwissenschaften* 55.4 (Dez. 2014), S. 139–153.

2 Vgl. „2650 Now Being Shipped And Sampled“. In: *Microcomputer Digest* 2.1 (Juli 1975). Hrsg. von Microcomputer Associates Inc., S. 1–3; Signetics Corporation, Hrsg. *Signetics 2650 Microprocessor*. Datenblatt. USA, 1975.

3 Das wohl bekannteste und erfolgreichste Signetics-Produkt ist der bis heute jährlich in Milliardenstückzahlen produzierte (und verkaufte) Timerchip NE555, vgl. Jack Ward. „The 555 Timer IC. An Interview with Hans Camenzind—The Designer of the Most Successful Integrated Circuit Ever Developed“. In: *Transistormuseum.com* (Juni 2004). URL: http://semiconductormuseum.com/Transistors/LectureHall/Camenzind/Camenzind_Index.htm (besucht am 24.09.2015), S. 1–2; Hans Camenzind. *Designing Analog Chips*. 2. Aufl. College Station/TX: Virtualbookworm Publishing, 2005. URL: http://www.designinganalogchips.com/_count/designinganalogchips.pdf (besucht am 24.09.2015), Kap. 11.

4 Vgl. Silicon Valley Historical Association. *NXP Semiconductors*. 2008. URL: <http://www.siliconvalleyhistorical.org/nxp-semiconductors-history> (besucht am 24.09.2015).

5 *Interton Electronic* war die Unterhaltungselektronik-Sparte des deutschen Hörgeräteherstellers *Interton*. Der *VC 4000 Video Computer* wurde ca. zwischen 1978 und 82 als lizenzierte Variante des britischen *Radiofin 1292 Advanced Programmable Video System* vertrieben. Historische Informationen zum *VC 4000* finden sich im Internet in einem rudimentären Wikipedia-Eintrag und auf den Seiten von Hobbyisten und Privatsammlern. Die dort zu findenden Angaben widersprechen sich jedoch zum Teil erheblich. Vgl. z. B. Eintrag „*Interton VC4000*“. In: *Wikipedia. Die freie Enzyklopädie*. Hrsg. von Wikimedia Foundation Inc. 15. März 2015. URL: https://de.wikipedia.org/w/index.php?title=Interton_VC4000&oldid=139806399 (besucht am 24.09.2015); Andreas Thurow. *Geschichte*. In: *VC4000.de*. Ohne Datierung. URL: <http://vc4000.de/html/>

ble Video Interface (PVI) 2636 als Coprozessor für Grafik und Klänge.⁶ Die Spielsoftware wird über programmierbare Speicher-ICs (sogenannte PROMs, *Programmable Read-Only Memory*) innerhalb von Spielmodulen geladen. Wiederbeschreibbarer Speicher (RAM, *Random Access Memory*) ist, abgesehen von 37 Bytes innerhalb des PVI, nicht vorhanden, es sei denn, das Spielmodul bringt entsprechende RAM-Bausteine mit. Die für die VC 4000 erhältlichen 37 Originalspiele variieren in ihrer möglichen Programmlänge aufgrund der verbauten ROM-Bausteine zwischen 2 KiB, 4 KiB und 6 KiB. Die Spiele „Backgammon“, „Dame“, „Schach 1“ und „Schach 2“ besitzen zusätzlich 1 KiB RAM-Speicher, z. B. für die Berechnungen der Züge des Computergegners oder zum Protokollieren der Spielzüge einer Schachpartie, die so nach Spielende nachvollzogen werden kann.⁷

Während des Workshops konnte ausschließlich in einer Emulationssoftware, dem von James Jacobs entwickelten *WinArcadia*, programmiert werden, die mehrere Heimcomputer- und Videospiele auf Basis des 2650 emuliert.⁸ Dank eingebautem Assembler und Speichermonitor sowie der Möglichkeit für die Kursteilnehmer, auf ihren eigenen Rechnern zu arbeiten, erschien dies zunächst sehr komfortabel, erwies sich letztendlich jedoch als äußerst problematisch. Als zum Abschluss des Workshops im September 2014 das halb entwickelte Spiel – eine Adaption des Smartphone-Spiels FlappyBird – auf der Konsole getestet werden sollte, wurden die Unterschiede zwischen der Emulationssoftware und der realen Hardware deutlich. Zwar war der programmierte Vogel genau wie im Emulator in den gewünschten Farben zu sehen und ließ sich auch ebenso steuern, andere Aspekte der Spiel-Demo verhielten sich auf der Originalhardware jedoch ganz anders als erwartet.

Die von Juliane Schütte entwickelte Musikbegleitung des Spiels war im Emulator noch stark verrauscht, nun überraschte die Konsole durch absolut störungsfreie Soundwiedergabe. Ganz anders bei der Anzeige des Punktestands: Im bisherigen Programmcode sollten hier die seit Beginn des Spiels verstrichenen Sekunden gezählt werden, was innerhalb der Emu-

geschichte.html (besucht am 24.09.2015); W. Forster. *Interton VC-4000 Video Computer*. Eintrag zur VC 4000 auf der Pong Picture Page. 27. Nov. 2005. URL: http://www.pong-picture-page.de/catalog/product_info.php?manufacturers_id=78&products_id=110 (besucht am 24.09.2015).

6 Vgl. Valvo, Hrsg. *Signetics Programmable Video Interface (PVI) 2636*. Datenblatt. Ohne Datierung.

7 Eine bebilderte Auflistung der Originalspiele findet sich bei Andreas Thurow. *Spiele*. In: VC4000.de. Ohne Datierung. URL: http://vc4000.de/html/body_spiele.html (besucht am 24.09.2015). 1 KiB (sprich: „ein Kibi-Byte“) steht für $1 \cdot 2^{10}$ Byte = 1024 Byte. Binärpräfixe wie *kibi* werden von der *International Electrotechnical Commission* (IEC), der internationalen Normungsorganisation für die Elektrotechnik in Genf, zur eindeutigen Unterscheidung von den üblichen *Dezimal*präfixen wie *kilo* empfohlen ($1 \text{ kB} = 1 \cdot 10^3$ Byte = 1000 Byte). Vgl. z. B. Markus Kuhn. *Standardized units for use in information technology or “What is a Megabyte ...?”* 19. Juli 1999. URL: <https://www.cl.cam.ac.uk/~mgk25/information-units.txt> (besucht am 24.09.2015).

8 James Jacobs. *WinArcadia*. Programm-Quellcode. Version 24.21. URL: <http://amigan.1emu.net/releases/WinArcadia-src.rar> (besucht am 24.09.2015).



Abbildung 1: Anzeigefehler im entwickelten Spiel auf der Originalhardware.

lation auch gelang. Auf der Konsole hingegen blieb der Zähler nach einer Sekunde wieder stehen. Zuletzt erstaunte der größtenteils schwarze Hintergrund auf dem Fernsehbildschirm, da ein dunkelblauer Hintergrund vorgesehen war. Dieser war nur am äußeren Rand des Bildschirms und in ein paar wenigen horizontal in der Bildmitte gelegenen Rechtecken zu sehen (siehe Abbildung 1). Auf Nachfrage reagierte der Emulatorentwickler Jacobs mit folgenden Hinweisen:

1. The sound issue will be fixed for the next version [of the emulator, J. M.].
2. The console is expecting SCORELT/SCORERT⁹ to be written during the frame, not during vertical blank; the emulator is more tolerant of when it is written.
3. Memory at \$1F80..\$1FAC holds the background grid definitions; the game needs to clear this at boot.¹⁰

Bei den naiven Programmierversuchen war die Workshopgruppe also gleich auf mehrere Punkte gestoßen, in denen sich der Emulator anders verhält als die reale Hardware. Interessanterweise lassen sich zwei verschiedene Ursachen für die beobachteten Divergenzen festmachen. Jens-Martin Loebel klassifiziert diese zwei Fehlerarten wie folgt: „Entweder wurde unterlassen (vergessen), etwas zu implementieren (‘error of omission’), oder etwas wurde falsch implementiert (‘error of commission’) [...]“. Bei „Community-Emulatoren“ seien Unterlassungsfehler häufiger, da deren Entwickler „oft nur auf unzureichende Spezifikationen zurückgreifen können.“¹¹

Punkt 1 stellt einen *error of commission* dar, folglich kann der Autor dessen Behebung für die nächste Programmversion ankündigen. Die Punkte 2 und 3 sind tatsächlich Unterlassungsfehler, die dem Programmierer allerdings durchaus bewusst sind, kann er doch klar erklären, warum sich das programmierte Spiel in diesen Fällen auf der realen Hardware anders verhält als in seinem Emulator. Mehr noch, im Falle von Punkt 3 verzichtet er bewusst darauf, den Implementierungsfehler – im Sinne einer *genaueren* oder *vollständigeren* Emula-

⁹ SCORELT und SCORERT sind die Mnemonics für die zwei Speicherzellen FC8₁₆ und FC9₁₆ im PVI, die den anzuzeigenden Punktstand bereithalten, vgl. James Jacobs. *Interton VC 4000 Coding Guide*. Allgemeine Hinweise zur Codestruktur von VC-4000-Spielen. Erstellt am 7. Feb 2007. Feb. 10, 2014. URL: <http://amigan.yatho.com/i-coding.txt> (visited on 09/24/2015).

¹⁰ James Jacobs. *The sound issue will be fixed*. Kommentar auf Facebook in der geschlossenen Gruppe „Signetics 2650“. 24. Sep. 2014. URL: https://www.facebook.com/groups/392123604238327/permalink/643611902422828/?comment_id=643746732409345&offset=0&total_comments=8&comment_tracking=%7B%22tn%22:%22R1%22%7D (besucht am 24.09.2015).

¹¹ Jens-Martin Loebel. *Lost in Translation. Leistungsfähigkeit, Einsatz und Grenzen von Emulatoren bei der Langzeitbewahrung digitaler multimedialer Objekte am Beispiel von Computerspielen*. Glückstadt: Hülsbusch, 2014, S. 146.

tion – zu beheben, die beobachtete „Translation Gap“ bleibt in diesen Fällen also bestehen.¹²

Die Gründe für bestehende (und bestehenbleibende) Unterlassungsfehler in Emulatoren liegen nicht ausschließlich an „fehlende[n] Systemspezifikationen und Missverständnissen“, auch wenn ein Blick in das Datenblatt des Grafikprozessors der *VC 4000* dies durchaus nahelegt.¹³ Tatsächlich findet sich auf den gesamten 14 Seiten kein expliziter Hinweis darauf, dass irgendein Speicherbereich des *2636* zur korrekten Inbetriebnahme initialisiert werden müsste. Vermutlich gingen die Autoren davon aus, dass ihre primäre Leserschaft ausgebildete Ingenieure und nicht Geisteswissenschaftler wären, sodass sie das im Falle unseres fehlerhaften Spielcodes entscheidende technische Detail, dass eine elektronische Speicherzelle nach dem Anlegen ihrer Versorgungsspannung einen undefinierten, nicht vorhersagbaren Zustand einnehmen kann, schlicht wegließen.

Warum der Emulator sich hier anders verhält, offenbart ein Blick in seinen Quellcode. Innerhalb der Funktion `interton_setmemmap()`, die den zur Emulation der Hardware benötigten Speicher vorbereitet, ist die folgende Schleife zu finden:¹⁴

```
1 for ( i = 0; i <= 0x7FFF; i++)  
2 {   memory[ i ] = 0;  
3 }
```

Hier wird der gesamte vom emulierten *2650* innerhalb der emulierten Konsole adressierbare Speicherbereich mit Nullen initialisiert. In der Softwareentwicklung mit maschinennahen Programmiersprachen ist dies – zur Vereinfachung des *debuggings*, der Eliminierung von Fehlern im Programmcode – die übliche Praxis, denn auch hier sind die physikalischen Eigenschaften von elektronischen Speicherzellen bzw. die Eigenschaften des darauf laufenden Betriebssystems bekannt: „Automatic variables for which there is no explicit initializer have undefined (i. e. garbage) values.“¹⁵

Mit den Worten Friedrich Kittlers ließe sich an dieser Stelle polemisieren: „Eine Maschine kann also zugleich weniger und mehr, als ihre Datenblätter zugeben.“¹⁶ Formuliert 1991, auf der Suche nach einem Ausweg aus dem „Protected Mode“ seines „No Name AT 386 [...] unter

¹² Mit dem Begriff „Translation Gap“ bezeichnet Loebel „alle Abweichungen [...] bei der Rezeption eines komplexen digitalen Artefakts innerhalb einer Emulationsumgebung [...] gegenüber [...] der Originalumgebung des Artefakts.“ Loebel, *Lost in Translation*, S. 105.

¹³ Vgl. Valvo, *Signetics Programmable Video Interface (PVI) 2636*. Diese Version des Datenblatts ist als Scan auf Jacobs' Server zu finden und wurde demnach sehr wahrscheinlich auch zur Entwicklung seines Emulators herangezogen.

¹⁴ Jacobs, *WinArcadia*, Datei `interton.c`, Zeilen 48–50.

¹⁵ Brian W. Kernighan und Dennis M. Ritchie. *The C Programming Language*. 2. Aufl. Englewood Cliffs/NJ: Prentice Hall, 1988, S. 40.

¹⁶ Friedrich Kittler. „Protected Mode“. In: Ders. *Draculas Vermächtnis. Technische Schriften*. Leipzig: Reclam, 1993, S. 208–224, hier S. 221.

Microsoft DOS 3.3“,¹⁷ diene Kittler dieser Satz als Vehikel, die Literaturwissenschaft aufzufordern, „auf einem strikt technischen Feld nach ähnlichen Verfahren vor[zu]gehen [...], wie es die Diskursanalyse Foucaults für Reden und Texte vorgeschlagen hat.“¹⁸ Der empirische Selbstversuch des fehlerhaften Spielcodes, der sowohl vermeintliche Lücken im Datenblatt des *PVI*, wie auch im Quellcode des verwendeten Emulators aufdeckte, könnte demnach als einfache Bestätigung der kittlerschen Polemik aufgefasst werden. Stattdessen kam die Idee auf, gemeinsam mit Mario Keller, einem weiteren Teilnehmer des Programmierworkshops, das Scheitern der Spielprogrammierung in Assembler aufgrund von „unzureichenden Spezifikationen“ nicht bloß als Beweis dafür zu nehmen, dass Kittlers Texte mithin lediglich noch „als *historische Quelle* und nicht als ‚Kittler‘ zu lesen“ seien.¹⁹ Stattdessen soll *Draculas Vermächtnis* nun erst recht beim (Vor-)Wort genommen werden. Dort heißt es: „Leser werden nicht erfahren, wie Schreibmaschinen zu bauen, Computerprogramme zu schreiben oder Schaltpläne zu lesen sind.“²⁰

2 Die Aufgabenstellung

Anstatt Datenblätter und Schaltpläne nur zu lesen, beschlossen Mario Keller und der Verfasser, die unbefriedigende Situation für heutige Programmierer an der alten *VC 4000* zu verbessern und selbst ein Stück Hardware zu entwerfen. Dieses sollte vornehmlich den jeweils rund zwanzig Minuten dauernden Löschvorgang eines EPROMs (*Erasable and Programmable Read-Only Memory*) durch UV-Lichtbestrahlung umgehen, der während des Assemblerworkshops nötig war, bevor der selbstprogrammierte Code auf der *VC 4000* für einen Testlauf geladen werden konnte. Mit der geplanten Platine im Modulschacht der Konsole sollte es möglich sein, den assemblierten Programmcode von einem modernen Flash-Speicher wie einer SD-Karte zu laden, anstatt von einem nicht oder nur schwer veränderbaren (EP)ROM.

Gleichzeitig würde ein solches „Flash-ROM-Modul“ oder „Multi-ROM-Modul“, wie der Arbeitstitel lautete, weitere Forschungsfragen aus dezidiert medienwissenschaftlichem Interesse nicht nur ermöglichen, sondern auch wesentlich erleichtern. Trivialerweise würde zunächst das händische Wechseln von Original-Spielmodulen entfallen, da alle ihre ROM-Inhalte, einmal ausgelesen und auf der SD-Karte abgelegt, weniger als 100 KiB Speicherplatz belegen. Etwas weniger banal wäre der Test von softwarekompatiblen Spielen technisch ver-

17 Kittler, „Protected Mode“, S. 208.

18 Ebd., S. 222.

19 Claus Pias. „Friedrich Kittler und der ‚Mißbrauch von Heeresgerät‘. Zur Situation eines Denkbildes 1964 – 1984 – 2014“. In: *Merkur. Deutsche Zeitschrift für europäisches Denken* 69.791 (Apr. 2015), S. 31–44, hier S. 34.

20 Friedrich Kittler. *Draculas Vermächtnis. Technische Schriften*. Leipzig: Reclam, 1993, S. 8.

wandter Konsolen wie der Radiofin 1292, deren Module jedoch aufgrund anderer Abmessungen des Kunststoffgehäuses nicht in den Modulschacht der *VC 4000* passen. Obendrein würde das Projekt die Gelegenheit geben, die Eignung des Retrocomputings als Methode der Medienarchäologie auszuloten (siehe Abschnitt 5).

Knapp zusammengefasst handelt es sich bei dem geplanten Multi-ROM-Modul um eine Hardware-Emulation aller Bauteile, die auf einem Spielmodul für die *VC 4000* vorkommen können und zur korrekten Wiedergabe des Spiels beitragen. Wie der Blick in ein beliebiges Spielmodul für die *VC 4000* zeigt, handelt es sich bei dieser Hardware in den meisten Fällen um nichts weiter als einen ROM-IC. Ausnahmen stellen lediglich die vier oben genannten Spiele dar, welche auch über RAM-Bausteine verfügen. Hier sind neben den ROM- und den RAM-ICs zusätzliche Logik-ICs verbaut, die je nach der gewünschten Speicheradresse entweder einen ROM- oder die RAM-Bausteine aktivieren (siehe Abbildung 2).

Offen gelassen wurde zu Projektbeginn die Frage, auf welche Art und Weise die Auswahl des zu ladenden Programmcodes erfolgen sollte. Als Königsweg erschien eine Menüsoftware, die nach dem Einschalten der Konsole zunächst geladen, die Dateinamen der auf der SD-Karte befindlichen Programme auf dem Fernsehbildschirm auflistet und dem Nutzer anschließend die Auswahl eines Spiel mittels der Bedienelemente auf dem Handcontroller der Konsole ermöglichen würde. Dies würde die Entwicklung von Assemblercode für den *2650* erfordern und damit einen direkten Anschluss an den Programmierworkshop bedeuten. Als alternative, einfachere Lösung kam die Idee auf, zusätzliche Auswahlelektronik direkt auf dem Modul anzubringen. Über mechanische Schalter und eine einfache Anzeige, z. B. in Form zweier Siebensegment-LED-Ziffern könnte gezielt eines von bis zu 100 (bei dezimaler) oder von bis zu 256 (bei hexadezimaler Darstellung) Programmen ausgewählt werden.

3 Der Lösungsweg

Aufgrund des – zumindest für Elektroniker – sehr komplexen Projekts wurde beschlossen, sich dem Projektziel schrittweise anzunähern, indem nacheinander einzelne, möglichst isoliert funktionstüchtige Teilschaltungen entwickelt würden. Ihre Zusammenführung würde sodann das fertige Multi-ROM-Modul ergeben.

3.1 Analyse des Hardwareaufbaus

In einem ersten Schritt musste der Hardwareaufbau der *VC 4000* näher untersucht werden. Es galt zu verstehen, wie die Verschaltung der wichtigsten ICs in der Konsole – nament-

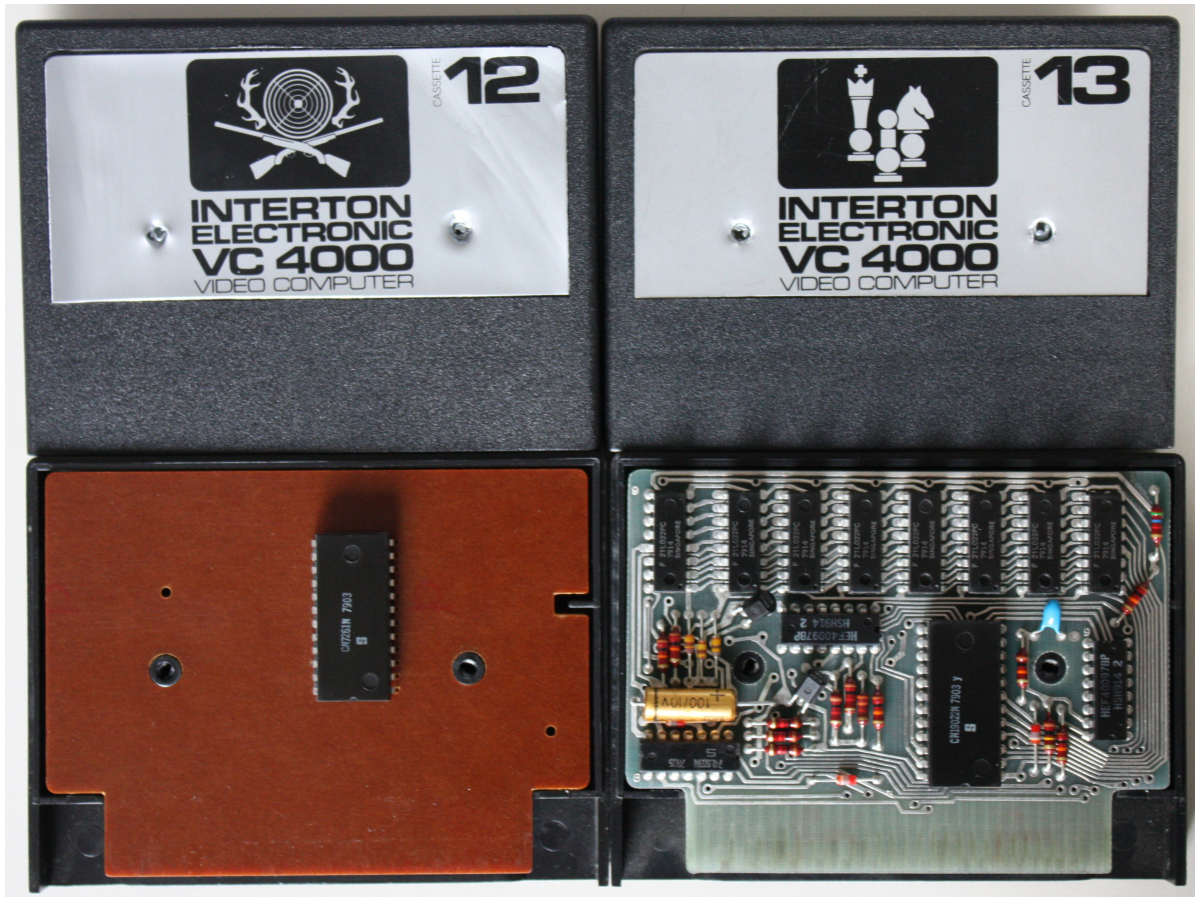


Abbildung 2: Zwei exemplarische Original-Spielmodule der VC 4000: links das Spielmodul Nr. 12 „Jagd“, das nur aus einem ROM-Baustein besteht; rechts das Spielmodul Nr. 13 „Schach“, das zusätzliche RAM-Bausteine und dafür nötige Steuerlogik mitbringt.

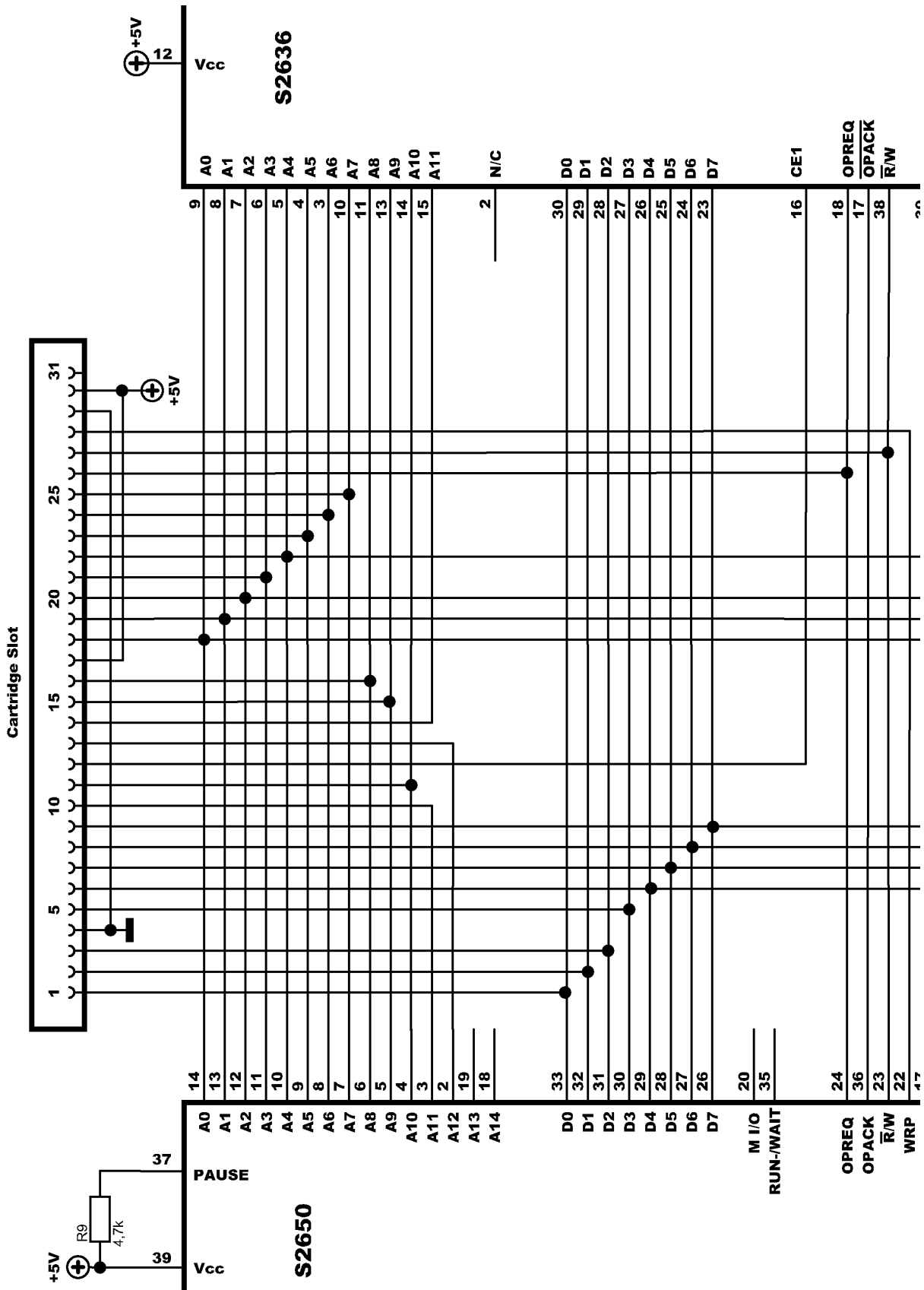


Abbildung 3: Die Verschaltung der CPU (S2650) und des PVI (S2636) mit dem Modulschacht.

A0 ... A12	Adressleitungen	Eingänge
D0 ... D7	Datenleitungen	Ein-/Ausgänge
CE1	Steuersignal ROM	Eingang
OPREQ, \bar{R}/W , WRP	Steuersignale RAM	Eingänge
+5 V, \perp	Betriebsspannung	Eingänge

Abbildung 4: Signalleitungen am Modulschacht. Der Signaltyp in der dritten Spalte ist aus der Perspektive des Spielmoduls angegeben.

lich die CPU 2650 und der PVI 2636 – mit den Bauteilen auf einem Spielmodul strukturiert ist. Hierbei konnte auf im Internet bereits verfügbare Schaltpläne zurückgegriffen werden.²¹ Der für das Projekt wichtigste Punkt der Schaltung ist zweifelsohne der Modulschacht oder *Cartridge Slot*, der die Schnittstelle zwischen der Konsole und dem zu entwickelnden Modul darstellt (siehe Abbildung 3).²² Die hierüber laufenden Leitungen transportieren diejenigen Signale, die die Konsole an ein Spielmodul richtet (eine 13 bit lange Adresse über den Adressbus) und als Antwort darauf erwartet (ein 8 bit langes Datenwort über den Datenbus), also diejenigen Hardwarefunktionen, die vom geplanten Multi-ROM-Modul verarbeitet bzw. ausgegeben werden müssen, um als Hardware-Emulation eines beliebigen Original-Spielmoduls zu funktionieren. Negativ formuliert sind die Signalleitungen am Modulschacht die einzigen Möglichkeiten, in den Funktionsablauf der Konsole einzugreifen (siehe Abbildung 4).

Diesem ersten Blick auf den Schaltplan folgte sogleich die erste Ernüchterung. Denn leider liegt am Modulschacht keine Leitung an, mit der das Multi-ROM-Modul die Signetics-CPU in ihrem laufenden Betrieb unterbrechen könnte, um beispielsweise – nach der Auswahl eines Spiels aus einer Menüsoftware heraus – die Konsole neu starten zu können, um das Spiel zu laden.²³ Jegliche Kommunikation zwischen Multi-ROM-Modul und Konsole muss demnach ausschließlich über den Adress- und den Datenbus geschehen.

Ausgerüstet mit diesen ersten Erkenntnissen über den Hardwareaufbau der Spielkonsole musste geklärt werden, welche Hardwarefunktionen das Multi-ROM-Modul besitzen muss und wie sich diese mit heutigen Bauteilen sinnvoll umsetzen lassen. Zu diesen Zeitpunkt waren Mario Keller und der Verfasser sich bereits darüber einig, dass mit der Mikrocontroller-Entwicklungsumgebung *Arduino* gearbeitet würde, sobald es komplexere Steuerungsaufga-

²¹ Die auf verschiedenen Servern innerhalb eines Komprimierten RAR-Archivs zu findenden Schaltpläne der VC 4000 stammen laut aufgedruckter Angabe allesamt von Manfred Schneider. *Schaltpläne der VC4000*. Archivdatei mit PDF-Zeichnungen. Feb. 2007. URL: http://vc4000.de/vc4000_Schaltplaene.rar (besucht am 24. 09. 2015).

²² Ebd., Detailausschnitt aus der Datei *vc4000-Mainboard.pdf*.

²³ Weder der RESET- (Anschluss 16) noch der INTREQ-Anschluss (*INTerrupt REQuest*, Anschluss 17) des 2650 werden zum Modul herausgeführt. Der RESET-Anschluss ist allerdings über den gleichnamigen Taster auf der Frontseite der Konsole zugänglich. Hier kann der Benutzer einen Neustart der CPU erzwingen.

ben auf der Hardware unseres Multi-ROM-Moduls zu realisieren gäbe. Diese Entscheidung hatte mehrere Gründe: Erstens hatten beide Projektteilnehmer bereits mehrere Projekte auf dieser Plattform realisiert, waren also mit ihrer Programmierung vertraut. Zweitens existieren für diese Plattform eine Vielzahl an fertigen Programmierbibliotheken (*libraries*) für bestimmte Anwendungen, die auch im Projekt benötigt würden. Zu nennen ist beispielsweise die SD-Library zum komfortablen Lesen und Schreiben von Daten von einer SD-Karte über Hochsprachenfunktionen, die auf der Arduino-Projektseite gut dokumentiert ist.²⁴ Drittens wurde die Arduino-Plattform eigens zu dem Zweck entwickelt, auch „nontechnical students“ die Programmierung von Microcontrollern zu Steuerungszwecken zu erleichtern.²⁵ Eine Arduino-Platine ist folglich Mittel der Wahl für technische Projekte, die im Rahmen eines nicht ingenieurwissenschaftlichen Studienganges wie der Medienwissenschaft entstehen.

Über den Mikrocontroller und die dafür verfügbaren Libraries hinaus kann der physische Anschluss einer SD-Karte ebenfalls über ein bereits fertig erhältliches, sogenanntes „Break-out-Board“ realisiert werden, das neben einem SD-Karten-Schacht zusätzlich über die entsprechenden Pegelumsetzer zwischen den in der *VC 4000* und auf dem Arduino herrschenden Spannungspegeln von 0 bis 5 V (den klassischen TTL-Pegeln) und den auf einer heutigen SD-Karte üblichen 0 bis 3,3 V verfügt. Durch solche Pegelumsetzer können ältere TTL-Schaltkreise gemeinsam mit moderneren Bauteilen in aktuellen Schaltungen verbaut werden, wie im Datenblatt des auf dem Breakout-Board verwendeten *SN74LVC125A* zu lesen ist: „Inputs can be driven from either 3.3-V or 5-V devices. This feature allows the use of this device as a translator in a mixed 3.3-V/5-V system environment.“²⁶

24 Vgl. Arduino, Hrsg. *Getting Started with Arduino*. Online-Dokumentation. URL: <https://www.arduino.cc/en/Guide/HomePage> (besucht am 24.09.2015). Sicherlich wäre die Programmierung des Zugriffs auf eine SD-Karte über das standardisierte „Serial Peripheral Interface“ (SPI) in Assembler der gangbare Weg für ein aus einem Assemblerworkshop entstandenes Projekt. Doch da das Programmierinterface zur SD-Karte, genauso wie andere Teilaspekte der Programmierung auf der Arduino-Plattform im Vergleich zum Projektziel eines Flash-ROM-Moduls für die *VC 4000* lediglich neben bzw. untergeordnete Rollen spielen, wird die Verwendung von Hochsprachenbibliotheken zur Pflicht. Andernfalls riskiert dieses Hardwareprojekt seine ganz eigene Softwarekrise. Eine Komplettentwicklung der Arduino-Software in Assembler umwillen maschinennaher Programmierung um jeden Preis würde das eigentlichen Projektziel torpedieren; „so sinnvoll dies gewesen sein mag, um grundlegende Arbeitsweisen des Computers zu begreifen“, um einen Kritikpunkt einer Medienwissenschaft „nach Kittler“ etwas zweckzuentfremden (Pias, „Friedrich Kittler und der ‚Mißbrauch von Heeresgerät‘“, S. 34). Nichtsdestotrotz ist der Quellcode all dieser Libraries im Umfeld der Arduino-Plattform dank Open-Source-Lizenzen jederzeit einsehbar. Open Source verschließt Hardwarefunktionen nur oberflächlich vor einem rein praxisorientierten Programmierer, nicht jedoch dem ebenso theoretisch interessierten Medienwissenschaftler. Zur historischen Softwarekrise Ende der 1960er Jahre, vgl. z. B. Michael S. Mahoney. „The histories of computing(s)“. In: *Interdisciplinary Science Reviews* 30.2 (2005), S. 119–135, hier S. 128.

25 David Kushner. „The Making of Arduino. How five friends engineered a small circuit board that’s taking the DIY world by storm“. In: *IEEE Spectrum* (26. Okt. 2011). URL: <http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino> (besucht am 24.09.2015).

26 Texas Instruments Inc., Hrsg. *SN74LVC125A. Quadruple Bus Buffer Gate With 3-State Outputs*. Datenblatt. Dallas/TX, Feb. 2004. URL: <http://pdf1.alldatasheet.com/datasheet-pdf/view/171748/TI/>

Sodann musste geklärt werden, wie der Programmcode der einzelnen Spiele für die Konsole bereitgestellt werden könnte, um spielbar zu sein. Wie aus Abbildung 4 ersichtlich, werden insgesamt 25 Signalleitungen benötigt. Dies schloss die alleinige Verwendung eines Arduinos aus, der in den einfachen Ausführungen lediglich über 20 frei programmierbare Anschlüsse verfügt.²⁷ Doch selbst wenn ein größerer Mikrocontroller wie auf einem Arduino Mega verwendet würde, der über genügend Anschlüsse verfügt, gäbe es eine strikt zeitkritische Grenze. Patrick Vogt, der ganz zu Beginn des Projekts einige Überlegungen für ein Multi-ROM-Modul über die Kollaborationsplattform *Moodle* beisteuerte, hat diese Grenze durch eine Überschlagsrechnung beziffert:

Selbst wenn der Mikrocontroller 20MHz²⁸ besitzt würde dies bedeuten, dass ein CLK [eine Taktperiode des Hauptprozessors, J.M.] $\sim 50\text{ns}$ entspricht. Da das gewünschte Byte, welches an den Adressleitungen anliegt nach ca. 450ns zurückgegeben werden muss, ergibt dies eine maximale Anzahl von 9 Befehlen (unter der idealisierten Annahme, das alle Befehle genau 1 CLK für die Verarbeitung benötigen).²⁹

3.2 Erste Hardwaretests mit einem SRAM

Statt des folglich unmöglichen Echtzeitmodus wurde beschlossen – in Anlehnung an einen weiteren Hinweis von Patrick Vogt aus dem soeben zitierten Forenbeitrag, in dem er vorschlug „mithilfe des Mikrocontrollers“ ein EEPROM (*Electrically Erasable Programmable Read-Only Memory*³⁰) mit dem gewünschten Programmcode zu brennen –, auf dem Multi-ROM-Modul einen dezidierten Speicherbaustein für die Bereitstellung des Spielcodes für die VC 4000 zu verwenden. Anstelle eines EEPROMs fiel die Entscheidung jedoch auf ein statisches RAM (SRAM).³¹ Neben dem Wegfall der Grenze physikalisch möglicher Schreib-

LVC125A.html (besucht am 24.09.2015), S. 1.

27 Von diesen 20 werden bereits zwei für das USB-Programmierinterface des Mikrocontrollers, sowie vier weitere für das SPI-Interface zur SD-Karte benötigt, sodass effektiv bloß 14 Anschlüsse für die Kommunikation mit der Konsole zur Verfügung stehen.

28 Die meisten Arduino-Platinen – z.B. der im Projekt verwendete Nano oder der Mega mit zusätzlichen Anschlüssen – besitzen standardmäßig einen 16 MHz-Quartz, sind also etwas langsamer getaktet.

29 Patrick Vogt. *Interton VC4000 Homebrew-EEPROM-Modul?* Beitrag im Online-Forum *circuit-board.de*. 10. Apr. 2014. URL: <http://circuit-board.de/forum/index.php/Thread/12776-Interton-VC4000-Homebrew-EEPROM-Modul/?postID=324054#post324054> (besucht am 24.09.2015), Orthographie aus dem Original übernommen.

30 EEPROMS sind mit den älteren EPROMS verwandt. Während letztere sich nur mittels UV-Licht löschen lassen, können EEPROMS nicht nur elektronisch programmiert, sondern auch elektronisch wieder gelöscht werden, was ihre Verwendung gegenüber EPROMS vereinfacht. Da die Anzahl an möglichen Schreibvorgängen, bevor ein EEPROM unbrauchbar wird, jedoch meist auf einige 100 000 begrenzt ist, schied dieser Speichertyp für das Projekt aus.

31 SRAM ist dadurch charakterisiert, dass die Speicherzellen – einmal beschrieben – ihren Speicherinhalt ohne äußeres Zutun behalten, solange eine Betriebsspannung anliegt. Die Bezeichnung SRAM wird vor allem

vorgänge ist diese Wahl auch durch eine überaus einfache elektronische Ansteuerung gerechtfertigt:

Using asynchronous SRAM is as easy as falling off a log: to READ a word, you assert the address, chip enable (CE') and output enable (OE'); The requested data obligingly appear on the three-state data lines a maximum of t_{AA} (address-access time) later.³² To WRITE a word, you assert address, data, and CE', then follow (after a minimum of an address setup time, t_{AS}) with a write enable (WE') pulse; valid data are latched into memory at the end of the WE' pulse.³³

Beim verwendeten SRAM-IC 6264 handelt es sich um einen 8 KiB (8192 Byte) großen Speicherbaustein, der genau wie die Konsole über einen 13 bit breiten Adressbus und einen Datenbus mit einer Wortbreite von 8 bit verfügt, die sich folglich direkt mit den entsprechenden Bussen der VC 4000 verschalten lassen.³⁴

Die beschränkte Ausgangszahl am Arduino zwang jedoch zur Verwendung weiterer Bausteine, um das SRAM beschreiben zu können. Durch die Hintereinanderreihung dreier sogenannter Schieberegister vom Typ 74HC595 werden am Arduino lediglich vier Ausgänge benötigt, um die Signale für einen Schreibvorgang – 13 Adress- und acht Datenbits – *seriell* auszugeben. Die Schieberegister geben die empfangenen Bits dann über ihre je acht Ausgänge *parallel* an die entsprechenden Adress- bzw. Datenanschlüsse am SRAM weiter, wo sie gespeichert werden sollen.³⁵ Wie von Horowitz und Hill prinzipiell beschrieben, ist

im Vergleich mit dynamischem RAM (DRAM) klar, wo der Speicherinhalt jeder einzelnen Speicherzelle in Zyklen wiederaufgefrischt werden muss, um nicht verloren zu gehen. SRAM-Zellen sind meist mittels Flipflop-Schaltungen aus sechs Transistoren realisiert, wohingegen eine DRAM-Zelle bereits mit nur einem Transistor und einem Kondensator implementiert werden kann – was DRAM-Bausteine zwar *deutlich* günstiger macht als vergleichbare SRAMs, jedoch den Nachteil einer für den geplanten Zweck zu komplizierten externen Wiederauffrischungsschaltung mit sich bringt. Für eine aktuelle, dezidiert elektrotechnische Diskussion mit entsprechenden Prinzipschaltungen, vgl. z. B. Paul Horowitz und Winfield Hill. *The Art of Electronics*. 3. Aufl. New York: Cambridge University Press, 2015, S. 1015–1021.

32 Zu den *three-state data lines*, siehe den folgenden Abschnitt.

33 Horowitz und Hill, *The Art of Electronics*, S. 1016 f.

34 Der 6264 wird als standardisierter SRAM-IC von verschiedenen Herstellern gebaut und angeboten, so z. B. von Utron Technology Inc., Hrsg. *UT6264C. 8K X 8 Bit Low Power CMOS SRAM*. 1.4. Datenblatt. Hsinchu, 15. Mai 2003. URL: <http://cdn-reicheit.de/documents/datenblatt/A300/UT6264.PDF> (besucht am 24. 09. 2015). Im Datenblatt werden als konkrete Werte für die minimale Dauer eines Lese, bzw. Schreibvorgangs 70 ns angegeben, vgl. ebd., S. 4.

35 Die serielle Übertragung der Daten vom Arduino und deren Übersetzung in parallele Signale für das SRAM funktioniert detailliert wie folgt: Über einen ersten Anschluss („SHCP, shift register clock input“) wird ein Taktsignal erzeugt, das den Schieberegistern signalisiert, in welchem Rhythmus an ihrem seriellen Eingang („DS, serial data input“) die entsprechenden Datenbits anliegen. Pro weiterem Taktsignal wird das so empfangene Bit um eine Stelle im internen Register weitergeschoben, während das nächste Bit am seriellen Eingang eintrifft. Sind alle Bits übertragen, wird über den dritten Anschluss namens „STCP“ („storage register clock input“) der aktuelle Inhalt des Schieberegisters in ein internes Speicherregister kopiert. Dessen Speicherzellen sind mit den acht parallelen Ausgängen des 74HC595 verbunden, die über den vierten Anschluss („OE, output enable input“) aktiviert werden können. Vgl. NXP Semiconductors, Hrsg. *74HC595; 74HCT595. 8-bit serial-in, serial or parallel-out shift register with output latches; 3-state*. 7. Rev. Datenblatt. 26. Jan. 2015. URL: <http://www.nxp.com/>

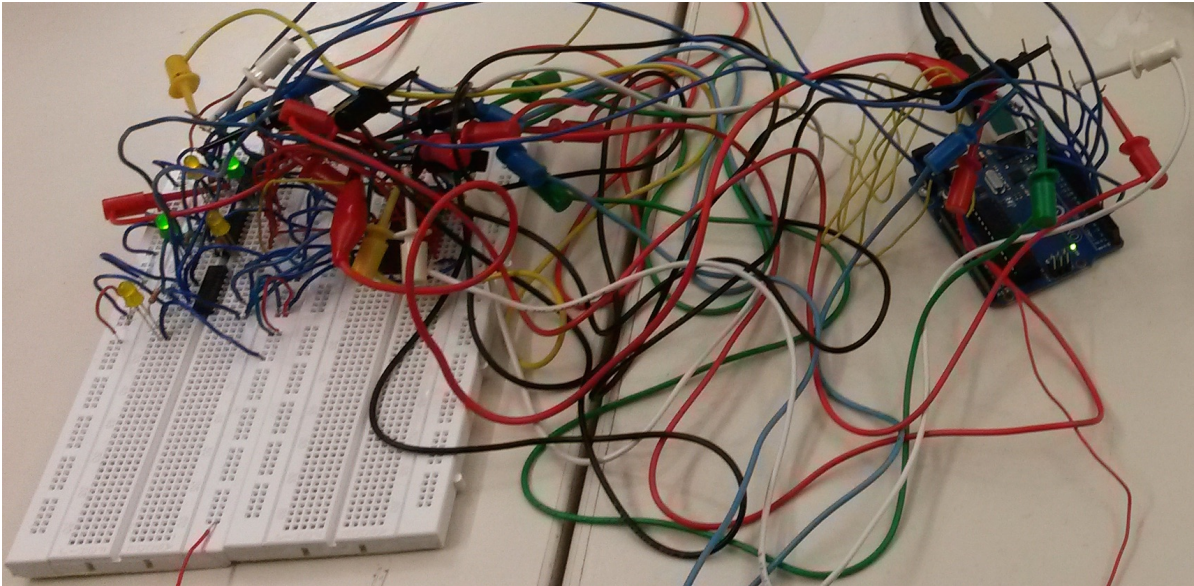


Abbildung 5: Testaufbau der seriellen Schreibschnittstelle zwischen SRAM und Arduino auf einem Steckbrett. Links die drei Schieberegister, rechts daneben der SRAM-IC (unter den Drähten), ganz rechts die Arduino-Platine.

dazu noch ein Schreibimpuls notwendig, der an einem weiteren Ausgang des Arduinos generiert werden kann. Zum Beschreiben des SRAMs werden am Arduino also insgesamt fünf Ausgänge benötigt.

Ein erster Test dieser Schreibschialtung mit den drei Schieberegistern, dem SRAM und einem Arduino UNO auf einem Steckbrett verlief im November 2014 erfolgreich (siehe Abbildung 5). Nach Abzug der fünf für die beschriebene serielle Schnittstelle benötigten Anschlüsse blieben am Arduino noch genügend Aus- und Eingangsleitungen übrig, um einen Speicherbereich von 256 Byte testweise wieder auszulesen. So konnte der Schreibvorgang verifiziert werden.

3.3 Der dritte Zustand binärer Elektronik

Es galt nun zu überlegen, wie diese Teilschialtung auf einem Multi-ROM-Modul innerhalb der VC 4000 eingesetzt werden kann. Denn unmittelbar nach dem Einschalten der Konsole beginnt nicht nur der Mikrocontroller des Arduinos seine Arbeit, sondern ebenso die CPU in der Konsole. Der 2650 legt – im Takt von etwa 900 kHz, das heißt rund einmal pro Mikrosekunde – Adresssignale an den Adressbus an und liest Datenbits vom Datenbus. Damit sind

documents/data_sheet/74HC_HCT595.pdf (besucht am 24.09.2015), S. 5, Tabellen 2 („Pin description“) und 3 („Function table“) sowie S. 6, Abbildung 8 („Timing diagram“).

alle diese Leitungen elektrisch belastet und der Arduino hätte keine Chance, sein Programm in den Speicher zu schreiben. Genausowenig würde die CPU ein ausführbares Programm vorfinden, denn die SRAM-Speicherzellen wären direkt nach dem Einschalten mit undefinierten Werten belegt.³⁶

Problematisch ist folglich, dass alle beteiligten Bauteile über dieselben elektrischen Leitungen miteinander verbunden sind und gleichzeitig versuchen, über diese Leitungen logische Binärwerte als Signalspannungen von entweder 0 oder 5 V zu übertragen. Im besten Fall passiert dabei nichts schlimmeres. Beide Prozessoren, der 2650 in der VC 4000 und der AVR ATmega 328 auf der Arduino-Platine, versuchen stur ihre Programme abzuarbeiten, was sich für den Benutzer am Fernsehbildschirm gegebenenfalls als visuelle und auditive Störungen (*glitch*) darstellen dürfte. Im schlechtesten Fall können jedoch durch die beteiligten Signalspannungen und dadurch ausgelöste Stromflüsse eines oder mehrere der beteiligten Geräte durch Kurzschlüsse zerstört werden.³⁷

Um dies zu verhindern, können die Ausgänge komplexerer elektronischer Digitalschaltungen in der Regel einen zusätzlichen *dritten* Zustand einnehmen, den sogenannten *hochohmigen* Zustand (auch *High-Z* oder schlicht *Z* genannt). Elektronisch betrachtet bedeutet dieser Zustand, dass eine Verbindung des Ausgangs mit den beiden Betriebsspannungspolen nur über sehr hohe elektrische Widerstände besteht, sodass über diesen Anschluss lediglich ein vernachlässigbarer Strom fließen kann.³⁸ Für die übrigen, an der selben Leitung angeschlossenen Geräte scheint es dann so, als wäre das Bauteil mit dem hochohmig geschalteten Ausgang gar nicht vorhanden.

Auf diese Weise funktioniert auch die normale Schaltung der VC 4000, wenn ein normales Spielmodul mit einem ROM-IC eingesteckt wurde. Denn bereits hier sind je drei Geräte mit dem Adress- und Datenbus verbunden: die CPU 2650, das PVI 2630 und das ROM auf dem Spielmodul, meist vom Typ 2616 oder 2632 (2 KiB, bzw. 4 KiB; siehe Abbildung 6).³⁹ Die Steuerung der Ausgänge D0–D7 des ROM erfolgt nicht nur über den Adressbus A0–A12, sondern zunächst über die Signale, die an den Eingängen mit den Bezeichnungen „CS“ (*Chip Select*),

³⁶ Genau wie die SRAM-Zellen innerhalb des PVI 2636, wie in Abschnitt 1 beschrieben.

³⁷ Vgl. z. B. die Warnung vor „Short circuits on Arduino Pins“ in Arduino, Hrsg. *Digital Pins*. Online-Dokumentation. URL: <https://www.arduino.cc/en/Tutorial/DigitalPins> (besucht am 24.09.2015), Abschnitt „Properties of Pins configured as OUTPUT“.

³⁸ Der stets existierende Reststrom (auch Leckstrom, *leakage current*, genannt) über hochohmige Anschlüsse ist laut den Datenblättern heutiger Digitalschaltungen normalerweise kleiner als 1 Mikroampere ($1 \mu\text{A} = 1 \cdot 10^{-6} \text{ A}$). Der entsprechende ohmsche Widerstand liegt dann im zweistelligen Megaohm-Bereich ($1 \text{ M}\Omega = 1 \cdot 10^6 \Omega$). Eingänge werden heutzutage im Regelfall immer als hochohmige Anschlüsse ausgelegt. Vgl. z. B. Philips Semiconductors, Hrsg. *74HC245; 74HCT245. Octal bus transceiver; 3-state*. 3. Rev. Datenblatt. 31. Jan. 2005. URL: http://www.nxp.com/documents/data_sheet/74HC_HCT245.pdf (besucht am 24.09.2015), S. 7, Tabelle 7, Symbole „ I_{LI} “, bzw. „ I_{OZ} “.

³⁹ Schneider, *Schaltpläne der VC4000*, Datei 2k-4krom-cartridge.pdf.

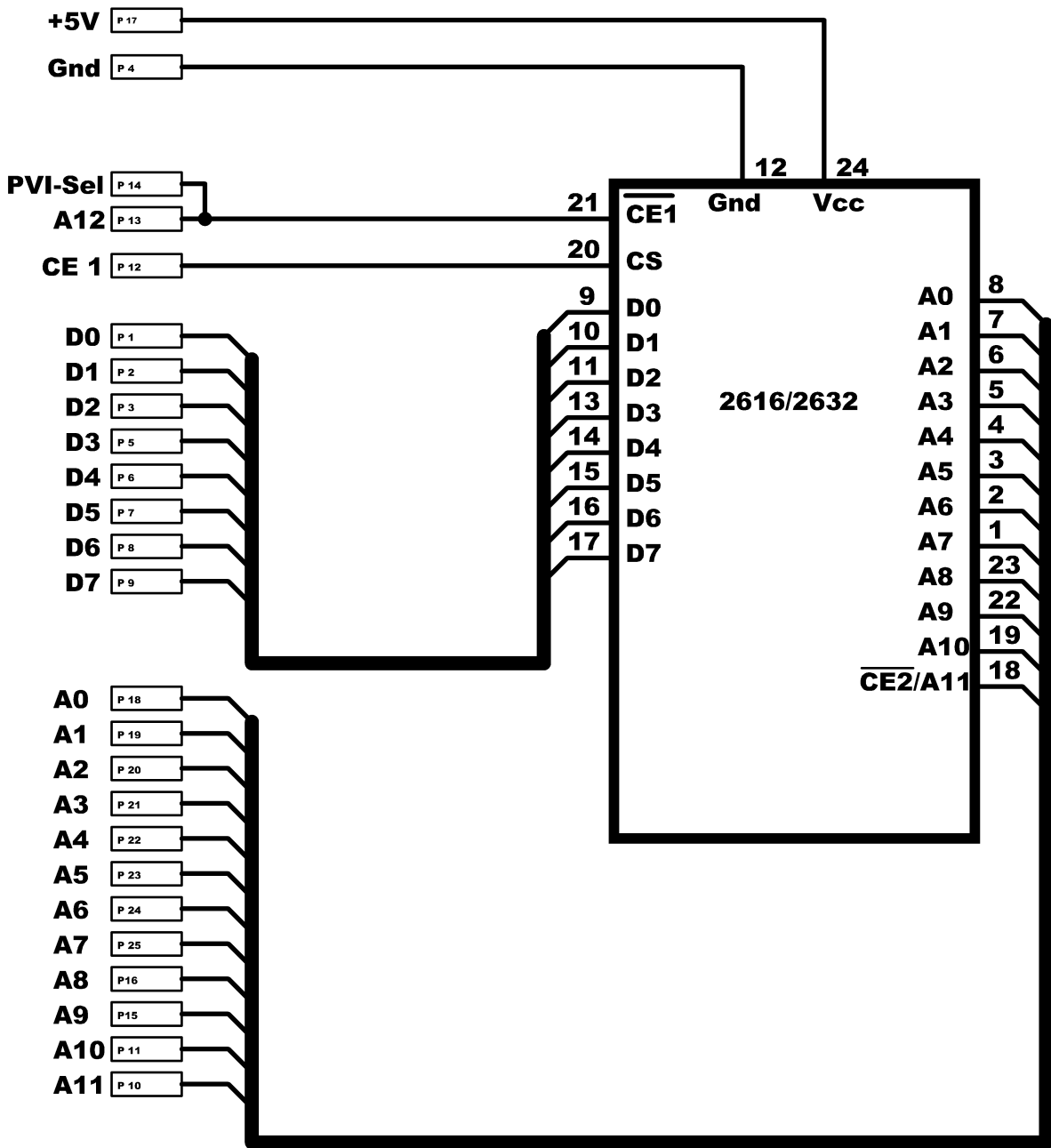


Abbildung 6: Schaltung auf einem Standard-Spielmodul für die VC 4000 mit ROM-IC.

bzw. „CE“ (*Chip Enable*) anliegen. Befindet sich bloß einer dieser Eingänge auf logisch-0 und ist demnach deaktiviert, so werden die Ausgänge in den hochohmigen Zustand geschaltet.⁴⁰ Wird zunächst nur die Leitung verfolgt, die am Anschluss 20 des ROM anliegt („CS“), so wird mit Blick auf Abbildung 3 deutlich, dass dieses Signal über Anschluss 12 am Modulschacht vom PVI kommt. Die Prosa im Datenblatt des 2636 bestätigt die diagrammatische Intuition der Schaltpläne:

The 2650A communicates with the PVI exclusively through memory addressing. [...] Two hundred and fifty-six addresses H'F00' to H'FFF' are used internally in the PVI as object and background description, scratchpad, and control. Accessing the first 3½K [H'000' to H'DFF', J. M.] will activate CE1—an active high output enable signal for external ROM.⁴¹

Liegt die von der CPU angeforderte Adresse im unteren Adressbereich, erzeugt der 2636 das „CE1“-Signal, welches am ROM-IC den „CS“-Eingang aktiviert, worauf dieses die der gewünschten Adresse entsprechenden Daten auf den Datenbus legt. Liegt die Adresse im Bereich oberhalb von DFF₁₆, muss der 2636 das gewünschte Byte aus seinem eigenen RAM-Bereich liefern. Dafür deaktiviert er die Spannung an seinem „CE1“-Ausgang, woraufhin der ROM-IC seine Ausgänge hochohmig schaltet. Innerhalb der VC 4000 entscheidet also der Grafik-Coprozessor, welches Gerät auf die gewünschte Adresse der CPU reagiert und folglich den Datenbus benutzen darf. Das jeweils andere Gerät muss seine Ausgänge im gleichen Takt hochohmig schalten.

Um im späteren Multi-ROM-Modul den Arduino in die Lage zu versetzen, das SRAM direkt nach dem Einschalten zunächst mit dem ausgewählten Spielcode zu beschreiben und dieses erst danach an die CPU der Konsole zu übergeben, kann nach dem gleichen Prinzip vorgegangen werden. Ein zusätzliches Bauteil wird zwischen Modulschacht und SRAM installiert, das sowohl Adresseingänge wie Datenleitungen zur Konsole hin hochohmig schaltet, solange der Arduino den Programmcode in das SRAM kopiert. Nach Abschluss des Schreibvorgangs werden die Leitungen zum Modulschacht wieder freigegeben. Wenn nun auf den RESET-Taster an der Konsole gedrückt wird, kann der 2650 bei Adresse 0 beginnend das eigentliche Spielprogramm aus dem SRAM auslesen, als wäre nun ein Spielmodul mit ROM-IC eingesteckt.

Ein solches Bauteil, das in der Lage ist, Ausgänge an einem Bussystem zu isolieren, wird

⁴⁰ Im Falle von „ $\overline{\text{CE1}}$ “ und „ $\overline{\text{CE2}}$ “ ist hierfür ein Eingangssignal von 5 V nötig, angezeigt durch die Überstreichung der Anschlussbezeichnungen, die beide Eingänge als *active low* deklariert. Das bedeutet, dass eine niedrige Eingangsspannung (0 V) hier eine logische 1 und eine hohe Eingangsspannung (5 V) eine logische 0 bedeutet, eine Invertierung der logischen Bedeutung der üblichen Spannungspegel.

⁴¹ Valvo, *Signetics Programmable Video Interface (PVI) 2636*, S. 3.

*Bus Transceiver*⁴² genannt und ist in der 74HC-Logikfamilie z. B. als 74HC245 erhältlich.⁴³ In einer solchen integrierten Schaltung sind acht einzelne Transceiver enthalten, deren elektronisches Verhalten über zwei Steuereingänge reguliert wird, „an output enable input (\overline{OE}) for easy cascading and a send/receive input (DIR) for direction control. \overline{OE} controls the outputs so that the buses are effectively isolated.“⁴⁴ Das Signal am DIR-Eingang (*direction*) legt fest, welche Seite aller acht Transceiver der Sender und welcher der Empfänger von Signalen auf der angeschlossenen Leitung ist. Mit dem \overline{OE} -Eingang können alle Leitungen auf der Senderseite bei Bedarf hochohmig geschaltet werden.

Benötigt werden drei 74HC245: Zwei sind für den Adressbus und die Steuereingänge CE1, OPREQ, $\overline{R/W}$ und WRP (siehe Abbildung 4) zuständig. Ihre DIR-Eingänge werden fest auf einen Spannungspegel eingestellt, sodass die Adresssignale von der Konsole stets die Empfängerseite der Bus Transceiver bilden. Die Leitungen, die vom Bus Transceiver weg zum SRAM und den Schieberegistern auf dem Multi-ROM-Modul führen, fungieren als Sender. Die Senderseite wird gleich nach dem Einschalten der Konsole vom Arduino über einen hohen Spannungspegel an \overline{OE} (logisch-0) hochohmig geschaltet. Somit gelangen die Adressanfragen vom 2650 nicht zum SRAM durch und der Arduino ist seinerseits in der Lage, die Programmdateien dort abzuspeichern.

Der dritte Bus Transceiver überwacht die acht Datenbus-Leitungen, wobei der DIR-Eingang im einfachsten Fall das im Vergleich zu den DIR-Eingängen an den Adressbus-Transceivern gegenteilige Signal erhält, sodass hier Sender und Empfängerseite vertauscht sind. Auf diese Weise wird das SRAM auf dem Multi-ROM-Modul zu einem Read-Only-Speicher gemacht, da der Datenbus so nur noch die vom SRAM kommenden Signale an die Konsole weitergibt. Mögliche Schreibversuche von der Konsole würden den Bus Transceiver blockieren. Auch am Datenbus-Transceiver wird über ein entsprechendes Signal an \overline{OE} gleich zu Beginn die Senderseite – diesmal zur Konsole hin – hochohmig geschaltet. Dadurch wird für die CPU und das PVI in der Konsole zunächst *ein leerer Modulschacht emuliert*. Für diese beiden Geräte *scheint es so*, als wäre kein Spielmodul eingesteckt, da auf ihrer Seite der Transceiver keine definierten Signale anliegen, während der Arduino auf dem Multi-ROM-Modul die Programmdateien in das SRAM kopiert.

42 Das englische Substantiv *transceiver* ist eine Kombination aus *transmitter* und *receiver*, kann also im Deutschen am ehesten mit Sender-Empfänger übersetzt werden. Siehe auch Abschnitt 5.3.

43 Die Abkürzung HC steht für *High-speed CMOS*, CMOS steht wiederum für *Complementary Metal-Oxide Semiconductor*. Anstatt Bipolartransistoren, wie in älteren 74er-Familien werden in CMOS-Technik schneller umschaltende und weniger Strom verbrauchende Feldeffekttransistoren integriert.

44 Philips Semiconductors, 74HC245, S. 1.

3.4 Auf der Jagd nach Hardware-Fehlern

Es wurde versucht, die soeben beschriebene Prinzipschaltung erneut auf einem Steckbrett zu testen. Dafür wurde eine weitere Arduino-Platine genutzt, um – nach dem Ende des Kopiervorgangs – auf der ‚Konsolenseite‘ der Bus Transceiver die Daten auf dem SRAM wieder auszulesen. Als dies erfolgreich gelang, wurde angenommen, es sei nun lediglich noch nötig, den Steckbrettaufbau über eine selbst hergestellte Breakoutplatine an den Modulschacht der VC 4000 anzuschließen, um ein erstes Spiel über den Modul-Prototypen auf der Konsole spielen zu können. Zu diesem Zweck wurde – mit derselben Arduino-Schaltung, mit der zuvor der Testaufbau überprüft wurde – der ROM-IC eines Originalspiels ausgelesen und dieser Code fest als zu kopierendes Spiel in den Arduino auf dem Multi-ROM-Prototypen einprogrammiert. Doch anstelle des erwarteten Jagdspiels erschien am Fernsehmonitor nur ein statischer, grauer Bildschirmhintergrund mit einem gelben Rahmen. An die Konsole angeschlossen, funktionierte etwas an der prinzipiellen Schaltung nicht, obwohl der Speicherinhalt des SRAMs nach dem Kopieren mit dem Inhalt des originalen ROM-ICs übereinstimmte.

Da sich während dieser Versuche ständig einige der Drahtverbindungen vom Steckbrett zum Modulschacht lösten, wurde nach einigen vergeblichen Stunden der Fehlersuche im März 2015 entschieden, für den bisherigen Schaltungsaufbau eine Platine herzustellen. So sollte sichergestellt werden, dass der beobachtete Fehler auch tatsächlich in der Schaltung lag und nicht in möglicherweise unzuverlässigen Drahtverbindungen auf dem Steckbrett. Mit dem freien Softwarepaket *KiCad* wurde zunächst ein klassischer diagrammatischer Schaltplan des Aufbaus erstellt, der sich mit einer solchen *Electronics Design Automation Suite* nach einigen Zwischenschritten in ein druckbares Platinenlayout übertragen lässt.⁴⁵ Dieses Drucklayout wird anschließend mittels eines UV-Beleuchters auf einem mit einer lichtempfindlichen Beschichtung versehenen Leiterplattenrohling abgebildet. Nach fotochemischer Entwicklung, die die gewünschten Leiterbahnen auf dem Rohling einschwärzt, werden die nicht benötigten Kupferflächen der Platine in einem Natriumpersulfat-Ätzbad entfernt. Nachdem die Löcher für die Bauteilanschlüsse gebohrt sind, können in einem letzten Schritt die Bauteile auf der Platine festgelötet werden.

Doch auch mit der nun gedruckten Schaltung unseres Testaufbaus trat das gleiche Problem

⁴⁵ Die nötigen Zwischenschritte zum Erstellen eines Leiterplattenlayouts aus einem Schaltplan sind: 1. Die Zuweisung von sogenannten „Bauteil-Fußabdrücken“ (*footprints*) zu den diagrammatischen Schaltsymbolen, sowie 2. das Erstellen einer Netzliste aus den Verbindungen der Bauteile dem auf Schaltplan. Diese Netzliste dient beim Zeichnen der Leiterbahnen zunächst als visuelle Hilfe und später, nach Vollendung des Entwurfs, zur automatischen Überprüfung auf Übereinstimmung mit dem Schaltplan durch das Layoutprogramm. Vgl. *KiCad. A Cross Platform and Open Source Electronics Design Automation Suite*. Website des verwendeten Platinenlayoutprogramms. URL: <http://kicad-pcb.org> (besucht am 24.09.2015), eine bebilderte Anleitung findet sich dort unter dem Link „Docs“.

										LOC	GRPA	GRPB
0000	1F00	2017	0620	0E47	A3CE	7F80	5A78	...		0	1F	0000
0010	F60E	26CA	CD7E	0A0F	26CA	CD7E	1AD9	...		1	00	0001
0020	20C3	CF5F	005B	7B92	CC5E	C9CC	1F0F	...		2	20	0002
0030	93BB	0404	17CC	1F0C	CC1F	1C06	013F	...		3	20	0003
0040	04FE	CC1F	2DCC	1F4D	0401	CC1F	5CCC	...		4	20	0020
0050	040F	CC1F	0E04	28CC	1F62	20CC	1F6C	...		5	C3	0021
0060	6DCC	1F6A	CC5E	6804	32CC	1F1E	0508	...		6	CF	0022
0070	C3CD	7FA0	5978	0414	CC1F	680D	1F5C	...		7	5F	0023
0080	9801	20CC	1F69	0D1F	6A18	0AA5	01E5	...		8	00	0624
0090	0904	081B	0204	04CC	1E80	CD1F	6A3F	...		9	FF	1FFF
										10	FF	0000
										11	5B	0027
										12	7B	0026
										13	79	0027
										14	CF	0022

Abbildung 7: Beginn des Spielcodes von „Jagd“: links ein Ausschnitt der aus dem Original-ROM-IC ausgelesenen Daten; rechts eine Bildschirmansicht der am Modulschacht gemessenen Signale, welche vom Logikanalysator in eine äquivalente Hexadezimaldarstellung übersetzt wurden. LOC zählt die laufenden Messungen, GRPA sind die Messungen am Datenbus, GRPB am Adressbus.⁴⁶

auf. Allerdings konnte nun mit ausreichender Sicherheit ausgeschlossen werden, dass beim Testaufbau auf dem Steckbrett ein Fehler gemacht wurde. Die rechnergestützte Bestätigung der Äquivalenz zwischen unserem diagrammatisch-logischen Schaltplan und seiner fotografisch-realen Übertragung auf die Leiterplatte grenzte die möglichen Fehlerquellen demnach weiter ein. Es wurde jetzt versucht, mittels zusätzlicher zeitkritischer Messungen durch einen Logikanalysator an der laufenden Konsole mit dem Prototypen im Modulschacht zu überprüfen, wann die tatsächlichen Signale vom erwarteten Verhalten abwichen.

Durch einen Logikanalysator können eine Vielzahl an gleichzeitig auftretenden Signalen in einer elektronischen Schaltung in ihrer zeitlichen Aufeinanderfolge gemessen und – automatisch in symbolische Werte übersetzt – auf ihre Korrektheit untersucht werden. Am Modulschacht der Konsole wurden acht Messleitungen an den Datenbus und weitere 13 an den Adressbus angebracht, wobei A0, die niederwertigste Adressleitung, als Taktsignal diente. Auf diese Weise war es möglich, zu verifizieren, dass die VC 4000, genau wie die Arduino-Testschaltung zuvor, zumindest den Beginn des Spielcodes korrekt aus dem SRAM auslas (siehe Abbildung 7). Dies bestätigte auch eine Vergleichsmessung mit dem Original-Spielmo-

⁴⁶ Ein paar Erläuterungen zur rechten Seite der Abbildung 7: In Messung 4 wurde der Sprung an Adresse 20_{16} vollzogen, in Messung 9 wird an Adresse $1FFF_{16}$ ins RAM des PVI geschrieben und in Messung 14 hat ein Rücksprung zu Zeile 22_{16} innerhalb einer programmierten Schleife stattgefunden. Die Abweichungen zwischen den Messungen 3, 9, 10 und 13 und dem Programmcode auf der linken Seite scheinen durch die CPU verursacht zu werden, die zur Ausführung einiger Instruktionen mehrere Takte benötigt oder für andere ihrerseits Daten auf den Datenbus legt. So sorgt z. B. der Speicherbefehl CF 5F 00 an Adresse 22_{16} dafür, dass die CPU den Wert FF_{16} an Adresse $1FFF_{16}$ in den Speicher schreibt, was in den Messungen 9 und 10 sichtbar ist.

dul, welche die gleichen Messergebnisse zeigte. Der Fehler konnte also nicht ausschließlich mit fehlerhaft aus dem SRAM ausgelesenen Werten des Programmcodes zu tun haben, denn andernfalls hätten die Messungen am Original-Spielmodul und am Prototypen gravierende Unterschiede aufweisen müssen. Mit Ausnahme einiger der Messungen, die während einiger der länger als zwei Takte dauernden Instruktionen gemacht wurden, ließen sich jedoch über den 256 Messwerte umfassenden Speicherbereich des Logikanalysators keine Unterschiede ausmachen.⁴⁷

Ob es sich bei den minimalen Messunterschieden tatsächlich um einen Hinweis auf das eigentliche Problem handelte oder ob die Signale, die während der zusätzlichen Takte bei Speicheroperationen ins PVI-RAM auf am Datenbus gemessen wurden, klassische Messfehler waren, wurde im Nachhinein nicht weiter überprüft. Festzuhalten ist jedoch, dass die unterschiedlichen Messwerte am Datenbus auf die richtige Fährte führten. Als im Folgenden die Schaltung des Bus Transceivers am Datenbus noch einmal genau überprüft wurde, konnte festgestellt werden, dass ein entscheidendes Detail bisher nicht beachtet war:

In der bisherigen Schaltung sorgte der Arduino dafür, dass die Ausgänge des Bus Transceivers am Datenbus nach dem Ende des Kopiervorgangs durchgehend aktiviert waren, der \overline{OE} -Eingang des *74HC245* lag permanent auf 0 V. Dies hatte zur Folge, dass der Datenbus durch das Multi-ROM-Modul elektronisch belastet war. Der Prototyp verletzte das erste Gebot der von-Neumann-Architektur: Zu jedem Zeitpunkt soll nur eine isolierte Operation durchgeführt – oder auf das vorliegende Problem umformuliert: nur ein Gerät darf zu jedem Zeitpunkt seine Signale auf einem Bus platzieren.⁴⁸ Auch bei Adresssignalen, die eigentlich nur den RAM-Bereich des PVI 2636 betreffen sollten, das dafür sein CE1-Signal deaktiviert hatte,⁴⁹ blieben die Ausgänge der Bus Transceiver aktiv. Mit dem so lokalisierten Fehler schien die Lösung trivial: Anstatt das CE1-Signal des PVI direkt auf den entsprechenden Eingang am SRAM zu legen, so wie es in Anlehnung an die Originalschaltung auf den Interton-Spielmodulen zunächst geplant war, musste dieses Signal lediglich auf den Anschluss 19 am Datenbus-Transceiver umgeleitet werden. Dieser kann dann den Datenbus hochohmig schalten, sobald eine Speicheroperation – egal ob lesend oder schreibend – im Bereich des PVI-RAMs (oberhalb von $1E00_{16}$) durchgeführt werden soll.

⁴⁷ Ein Beispiel für Abweichungen der Messungen vom Originalcode ist auch in Abbildung 7 zu sehen. Siehe dazu die Erläuterungen in Fußnote 46.

⁴⁸ Die entsprechende Stelle in von Neumanns „First Draft of a Report on the EDVAC“ bespricht zwar arithmetische Operationen in der CPU, diese schließen aber die Leitungen zum angeschlossenen Speicher mit ein: „The device should be as simple as possible, [...]. This can be achieved by never performing two operations simultaneously, [...]“. John von Neumann. „First Draft of a Report on the EDVAC“. In: *IEEE Annals of the History of Computing* 15.3 (1993), S. 27–75, hier S. 40, Abschnitt 5.6.

⁴⁹ Siehe Abschnitt 3.3.

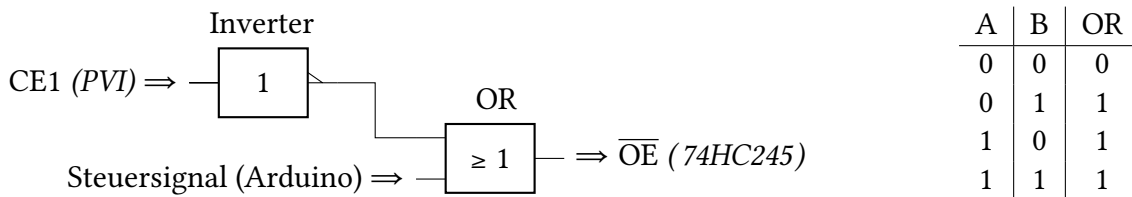


Abbildung 8: Verschaltung des *Output-Enable*-Eingangs am Datenbus-Transceiver; Wahrheitstabelle für ein OR-Gatter.

3.5 Zusätzliche Logikgatter

Selbstverständlich war der besagte Anschluss 19 – der *Output Enable* des 74HC245 am Datenbus – bereits belegt, nämlich mit dem in Abschnitt 3.3 beschriebenen Steuersignal vom Arduino, das unser Multi-ROM-Modul für die Vorbereitung des SRAMs von der Konsole isolieren sollte. Obendrein handelte es sich beim CE1-Signal vom PVI um ein *Active-High*-Signal, der Bus Transceiver erwartete jedoch ein *Active-Low*-Signal an seinem \overline{OE} -Eingang. Solche Probleme von zueinander inkompatiblen Signalen oder mehrfach zu belegenden Ein- und Ausgängen sind in der Digitalelektronik an der Tagesordnung. Lösungen liefern sogenannte Gatter, welche die logischen Grundfunktionen – AND, OR, XOR, NOT, NAND, NOR, XNOR usw. – in standardisierten integrierten Schaltungen anbieten.

Für das konkret beschriebene Problem lässt sich mit zwei Gattertypen aus der berühmten 74xx-Logikfamilie eine Lösung konstruieren (siehe Abbildung 8).⁵⁰ Zunächst wird das vom CE1-Ausgang des PVI kommende Signal *invertiert*, um es zum \overline{OE} -Eingang kompatibel zu machen. Dieses negierte Signal wird zusammen mit dem Steuersignal vom Arduino auf die Eingänge eines OR-Gatters gelegt. Die darin implementierte logische Funktion sorgt dafür, dass der Ausgang dieses Gatters auf einem hohen Spannungspegel liegt, sobald mindestens eines der beiden Eingangssignale aktiv ist.⁵¹ In der 74xx-Logikfamilie gibt es unter der Kennnummer 74HC32 vier OR-Gatter in einem Baustein, für den Inverter wurde eines

⁵⁰ Eine praxisorientierte Einführung in die 74xx-Logikfamilie aus elektronischer Sicht liefert Don Lancaster. *TTL Cookbook*. 25. Aufl. [Erstauflage 1974]. Carmel/IN: SAMS, 1994.

⁵¹ Die Funktionsweisen logischer Gatter spiegeln sich in den verwendeten Schaltsymbolen wieder: Die Kennzeichnung „ ≥ 1 “ im OR-Gatter benennt diejenige Anzahl an Eingangssignalen, die logisch-1 sein müssen, damit der Ausgang ebenfalls auf 1 schaltet. Analog verhält es sich mit dem Symbol des Inverters, der mit einem sogenannten „Puffer“ verwandt ist: Ein Puffer gibt das anliegende Eingangssignal unverändert weiter, implementiert also die logische Funktion der Identität. Dies wird durch eine „1“ symbolisiert: Das Ausgangssignal eines Puffers ist dann 1, wenn auch sein Eingangssignal 1 ist. Die Identitätsfunktion ist das logische Gegenteil der Negation, welche durch einen Inverter implementiert werden kann. Ein Gatter, das die komplementäre Funktion eines anderen Gatters implementiert, wird durch das gleiche Schaltsymbol dargestellt, mit einem zusätzlichen fallenden Strich über seinem Ausgang. Diese Schaltsymbole, die logische Gatter über die Verhältnisse von Ein- und Ausgangssignalen zueinander darstellen, wurde Ende 1997 von der IEC im Standard 60617-12 eingeführt.

der vier NAND-Gatter in einem *74HC00* genutzt.⁵² Ein NAND-Gatter lässt sich als Inverter einsetzen, indem das Eingangssignal auf beide Eingänge gelegt wird. Denn wenn beide Eingänge eines NAND-Gatters den gleichen logischen Wert haben, ergibt sich am Ausgang das jeweils gegenteilige Signal (siehe auch Abbildung 10). Durch diese zusätzliche Logik kann der Datenbus zur Konsole nicht nur kurz nach dem Einschalten durch den Arduino, sondern auch während des Spielbetriebs durch den PVI deaktiviert werden, wenn dessen interner RAM-Bereich adressiert wird.

3.6 (SD-)Kartenspiele

Mit dieser Modifikation gelang Ende Juli 2015 der Durchbruch. Es konnte erstmals ein Spiel direkt mit dem Prototypen auf der *VC 4000* gespielt werden (siehe Abbildung 9). Im Anschluss wurde der Firmwarecode auf dem Arduino modifiziert, um den Spielcode wie geplant von einer MicroSD-Karte laden zu können, auf der zuvor die ROM-Daten von allen ausgelesenen Spielmodulen abgelegt wurden. Da der Prototyp zu diesem Zeitpunkt noch nicht über Siebensegmentanzeigen und Auswahlaster verfügte, wurde pragmatisch ein aus Nutzersicht zunächst etwas umständliches, aber funktionierendes Verfahren implementiert.

Auf der SD-Karte musste eine zusätzliche Textdatei angelegt werden, in die der Nutzer den Dateinamen der zu ladenden Binärcodedatei einträgt. Der Arduino liest zunächst den Inhalt dieser Textdatei aus und sucht anschließend nach einer ROM-Datei mit dem entsprechenden Namen. Bei Erfolg wird deren Inhalt in das SRAM kopiert. Das Ende dieses Kopiervorgangs wird dem Nutzer durch regelmäßiges, langsames Blinken der auf der Platine angebrachten LED angezeigt, sodass durch Druck auf die RESET-Taste der Konsole das Spiel gestartet werden kann. Fehler in diesem sequentiellen Programmablauf, etwa eine nicht eingesteckte SD-Karte, eine fehlende Textdatei oder eine nicht vorhandene ROM-Datei, werden durch einen abweichenden Blinkrhythmus angezeigt.⁵³

Mit diesen Änderungen konnten alle Spielmodule, die ausschließlich über einen ROM-IC verfügen, erfolgreich über den Multi-ROM-Prototypen von SD-Karte geladen und gespielt werden.⁵⁴ Es blieben lediglich vier Spiele übrig, für die noch weitere Modifikationen an der

52 Vgl. NXP Semiconductors, Hrsg. *74HC32; 74HCT32. Quad 2-input OR gate*. 5. Rev. Datenblatt. 4. Sep. 2012. URL: http://www.nxp.com/documents/data_sheet/74HC_HCT32.pdf (besucht am 24.09.2015); NXP Semiconductors, Hrsg. *74HC00; 74HCT00. Quad 2-input NAND gate*. 6. Rev. Datenblatt. 14. Dez. 2011. URL: http://www.nxp.com/documents/data_sheet/74HC_HCT00.pdf (besucht am 24.09.2015).

53 Dieser Entwicklungsstand der Firmware kann – wie auch alle anderen hier im Text beschriebenen Hard- und Softwaremodifikationen – im Github-Repository des Multi-ROM-Projekts eingesehen werden: Mario Keller und Johannes Maibaum. *VC4000 MultiRom Modul*. Code-Repository. URL: <https://github.com/mkeller0815/flappy4000/tree/master/MultiRom> (besucht am 24.09.2015), Commit 5a9af7a vom 21.08.2015, Datei `MultiRom/firmware/MultiRomVC4000/MultiRomVC4000.ino`, Zeilen 37–71.

54 Eine Statusübersicht über die Unterstützung der verschiedenen Spiele durch das Multi-ROM-Modul findet

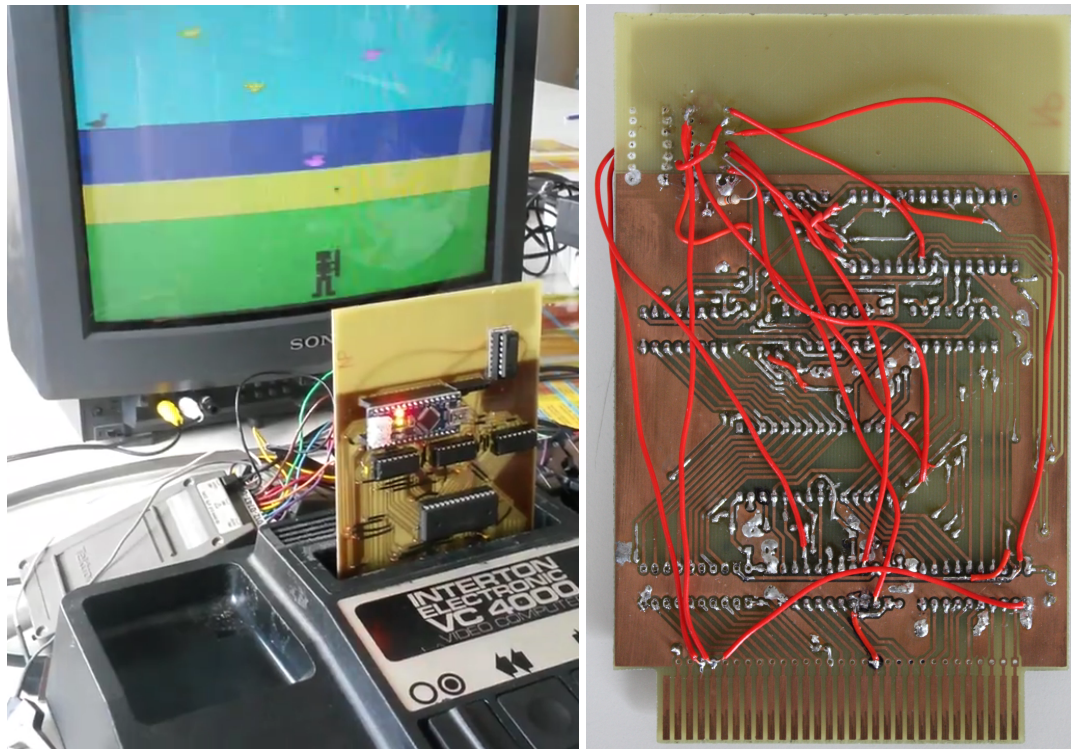


Abbildung 9: Der erste funktionierende Prototyp: links in Aktion mit dem Spiel „Jagd“, rechts ein Blick auf die Platinenrückseite mit den manuellen Änderungen.

Hardware nötig waren: „Schach“ (Nr. 13), „Schach 2“ (Nr. 22), „Dame“ (Nr. 26) und Backgammon (Nr. 36), die vier Spiele, die zusätzlichen RAM-Speicher auf dem Spielmodul mitbringen.

3.7 Lesen und Schreiben

Bis hierhin war der SRAM-IC auf dem Prototypen faktisch zu einem ROM degradiert, denn die vorgeschalteten Bus Transceiver ließen auf dem Datenbus nur Daten aus Richtung des SRAMs durch.⁵⁵ Um nun die genannten RAM-Spiele ebenfalls über das Multi-ROM-Modul spielen zu können, musste dieser Zustand geändert werden. Die hierzu nötigen Modifikationen hielten sich in Grenzen, da der RAM-Speicher auf den genannten Spielmodulen ebenfalls durch SRAM-ICs realisiert wurde. Ein Blick in das Datenblatt des dort häufig verwendeten 2114-SRAMs zeigte,⁵⁶ dass deren Ansteuerung genauso funktioniert, wie oben in Ab-

sich ebenfalls im Github-Repository, Datei MultiRom/doc/README.md.

⁵⁵ Alle DIR-Anschlüsse (*direction*) der drei Bus Transceiver waren zu diesem Entwicklungszeitpunkt fest auf einen logischen Spannungspegel gelegt, vgl. Keller und Maibaum, *VC4000 MultiRom Modul*, Commit ba2324a3e1 vom 28.07.2015, Datei MultiRom/hardware/Printouts/MultiRomCard-Schematic.pdf.

⁵⁶ MOS Technology Inc., Hrsg. *F2114/2114L. 1024 x 4 Static RAM*. Datenblatt. Ohne Datierung. URL: <https://www.princeton.edu/~mae412/HANDOUTS/Datasheets/2114.pdf> (besucht am 24.09.2015). Das

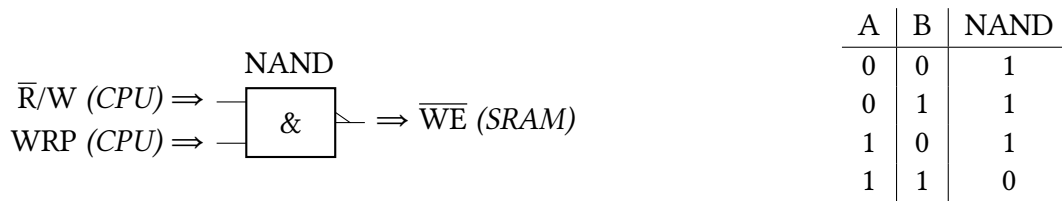


Abbildung 10: Erzeugung eines Schreibsignals für den \overline{WE} -Eingang des SRAMs; Wahrheitstabelle für ein NAND-Gatter

schnitt 3.2 bereits für das gewählte 6264-SRAM beschrieben. Demnach lieferte der Schaltplan der entsprechenden Spielmodule den Hinweis, wie der *Write-Enable*-Eingang (\overline{WE}) am SRAM durch die Signale von der VC 4000 zu beschalten ist, um einen erfolgreichen Schreibvorgang einzuleiten:⁵⁷ Die von der Konsole kommenden Signale \overline{R}/W und WRP werden dort über ein NAND-Gatter verschaltet (siehe Abbildung 10).

Im Datenblatt des 2650 findet sich das Zeitablaufdiagramm, das verdeutlicht, auf welche Art und Weise die CPU mittels der beiden genannten Signale einen Schreibvorgang orchestriert:⁵⁸ Am Beginn eines Zyklus wird \overline{R}/W auf hohen und WRP auf niedrigen Spannungspegel geschaltet. Zu diesem Zeitpunkt liegen die Speicheradresse und die zu schreibenden Daten bereits auf den jeweiligen Bussen. Nach einem „internal delay“ von 300 ns wird das WRP-Signal kurzzeitig eingeschaltet, bevor der Schreibzyklus durch die nächste fallende CLOCK-Signalfanke beendet wird. Nach einer weiteren Wartezeit von 500 ns kehren \overline{R}/W und WRP in ihre Ausgangszustände zurück (siehe Abbildung 11).

Das NAND-Gatter fungiert auf den Spielmodulen folglich als Übersetzer der – für die genutzten SRAM-ICs – inkompatiblen Steuersignale der CPU. Beide Signale vom 2650 werden für einen Schreibvorgang auf den hohen Spannungspegel gelegt, der \overline{WE} -Eingang am SRAM ist jedoch wiederum *active low*, für einen Speichervorgang wird hier ein niedriger Spannungspegel erwartet. Mit \overline{R}/W und WRP auf den Eingängen eines NAND-Gatters wird dieser niedrige Pegel genau dann erzeugt, wenn beide Signale ihrerseits im Zustand logisch-1 sind, also genau in der Mitte eines Schreibzyklusses. In allen anderen Fällen erzeugt das NAND-Gatter eine Signalspannung von 0 V.

Mit den Erfahrungen der letzten Hardwarefehlersuche im Hinterkopf, dass nicht vergessen werden darf, die Bus Transceiver in der Schaltung entsprechend mit zu berücksichtigen, wurde das \overline{R}/W -Signal, das über die gesamte Dauer eines Schreibzyklus bei 0 V liegt, au-

Foto vom Spielmodul Nr. 13 in Abbildung 2 zeigt, dass anstatt zweier 2114 manchmal auch acht SRAM-ICs vom Typ 2102 (1 KiB x 1 Bit) verbaut wurden.

⁵⁷ Schneider, *Schaltpläne der VC4000*, Datei 4krom-1kram-cartridge.pdf.

⁵⁸ Signetics Corporation, *Signetics 2650 Microprocessor*, S. 36, Abb. 11. \overline{R}/W wird dort als $\overline{READ}/WRITE$ ausgezeichnet.

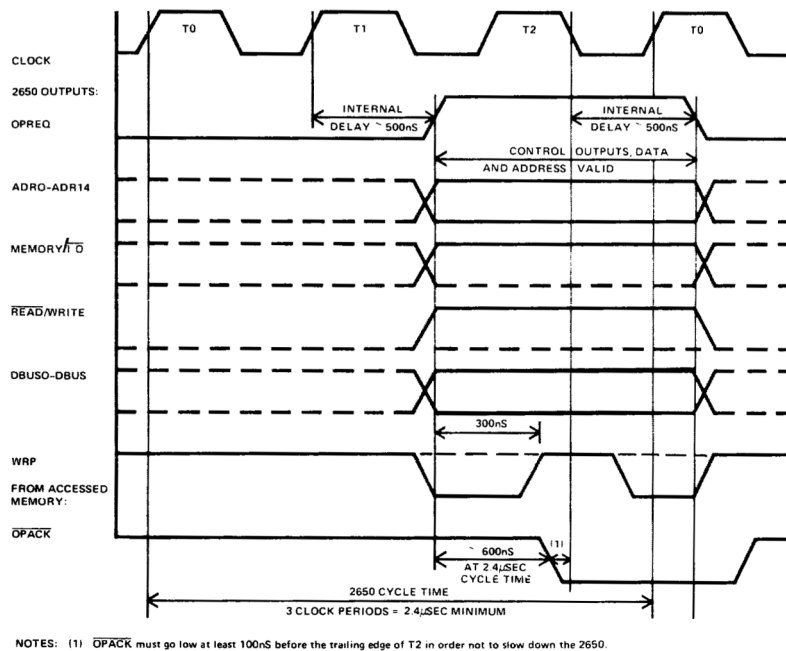


Abbildung 11: Schreibzyklus des *Signetics 2650*.

ßerdem auf den DIR-Anschluss des Datenbustransceivers gelegt. Dieses Signal stimmt passenderweise mit der für die entsprechende Richtungsänderung des *74HC245* notwendigen Spannung überein, sodass hier keine weitere Logikschaltung nötig war. Bei einem positiven $\overline{R/W}$ -Signal, das den Schreibvorgang einleitet, lässt der Transceiver die Datenbussignale von der CPU in Richtung des Multi-ROM-Moduls passieren; nach Ende des Schreibvorgangs, wenn $\overline{R/W}$ fällt, schaltet somit auch der Bus Transceiver wieder um, sodass die nächste Leseoperation aus dem SRAM gelingen kann.

3.8 Das letzte fehlerhafte Spiel

Nach diesen Änderungen ließen sich alle Spielmodule für die *VC 4000* auf dem Multi-ROM-Prototypen spielen, mit Ausnahme von „Schach 2“. Das einzige 6 KiB-ROM- und 1 KiB-RAM-Spiel ließ sich zwar erfolgreich laden und man konnte ein paar Züge gegen den Computergegner spielen. Nach einiger Zeit reagierte die Konsole jedoch nicht mehr auf Eingaben am Controller oder der errechnete Gegenzug, angekündigt durch ein kurzes Piepen der Konsole, wurde nicht mehr auf dem Bildschirm angezeigt. Es wurde angenommen, dass diese Fehler erneut durch den Hardwareaufbau verursacht würden. Denn bisher nicht berücksichtigt wurde die Tatsache, dass die Art der Verschaltung der SRAMs auf den originalen Spielmodulen dafür sorgt, dass Teile ihres Speicherbereichs von zwei unterschiedlichen Adressberei-

chen heraus aufgerufen werden können.

Wie aus Abbildung 12 ersichtlich wird, besitzen die beiden 2114-SRAMs je nur zehn Adressleitungen (A0–A9), was der angegebenen Speichergröße von 1 KiB entspricht. Wie weiterhin aus der Wahrheitstabelle in der unteren rechten Ecke des Schaltplans hervorgeht, ist ihr *Chip-Enable*-Anschluss jedoch über den Adressbereich von 1800_{16} – $1DFF_{16}$ aktiv. Das bedeutet, dass die SRAM-ICs über insgesamt 1536 Adressen, anstatt der nominellen 1024 angesprochen werden können.⁵⁹ Doch welche Speicherzellen innerhalb der SRAMs liefern die Daten, wenn Adresse $1C00_{16}$ (die 1025. Adresse des genannten Bereichs) angefordert wird? Dies wird deutlich, wenn die hexadezimalen Adressangaben im dualen Zahlensystem dargestellt werden, sodass jedes Bit einer tatsächlichen Adressleitung entspricht (siehe Abbildung 13). Bis auf das elfte Bit von rechts, das A10 entspricht, sind die beiden Adressen 1800_{16} und $1C00_{16}$ identisch. Den SRAM-ICs fehlt dieser Anschluss, demnach steuern lediglich die unteren zehn Adressleitungen (A9–A10) den SRAM-Speicher an. Da diese in beiden Fällen übereinstimmen, wird auch dieselbe Speicherzelle ausgewählt. Dies gilt für alle Adressen oberhalb von $1C00_{16}$, was in der technischen Informatik auch als „Adressspiegelung“ (*mirroring*) bezeichnet wird.⁶⁰ Dass im Spielmodul zu „Schach 2“ nicht der gesamte RAM-Bereich gespiegelt wird, liegt am CE1-Signal vom PVI, das ab Adresse $1E00_{16}$ alle Speicher-ICs auf dem Spielmodul deaktiviert, um den Zugriff auf seinen internen Speicherbereich zu ermöglichen (siehe auch Abschnitt 3.3).

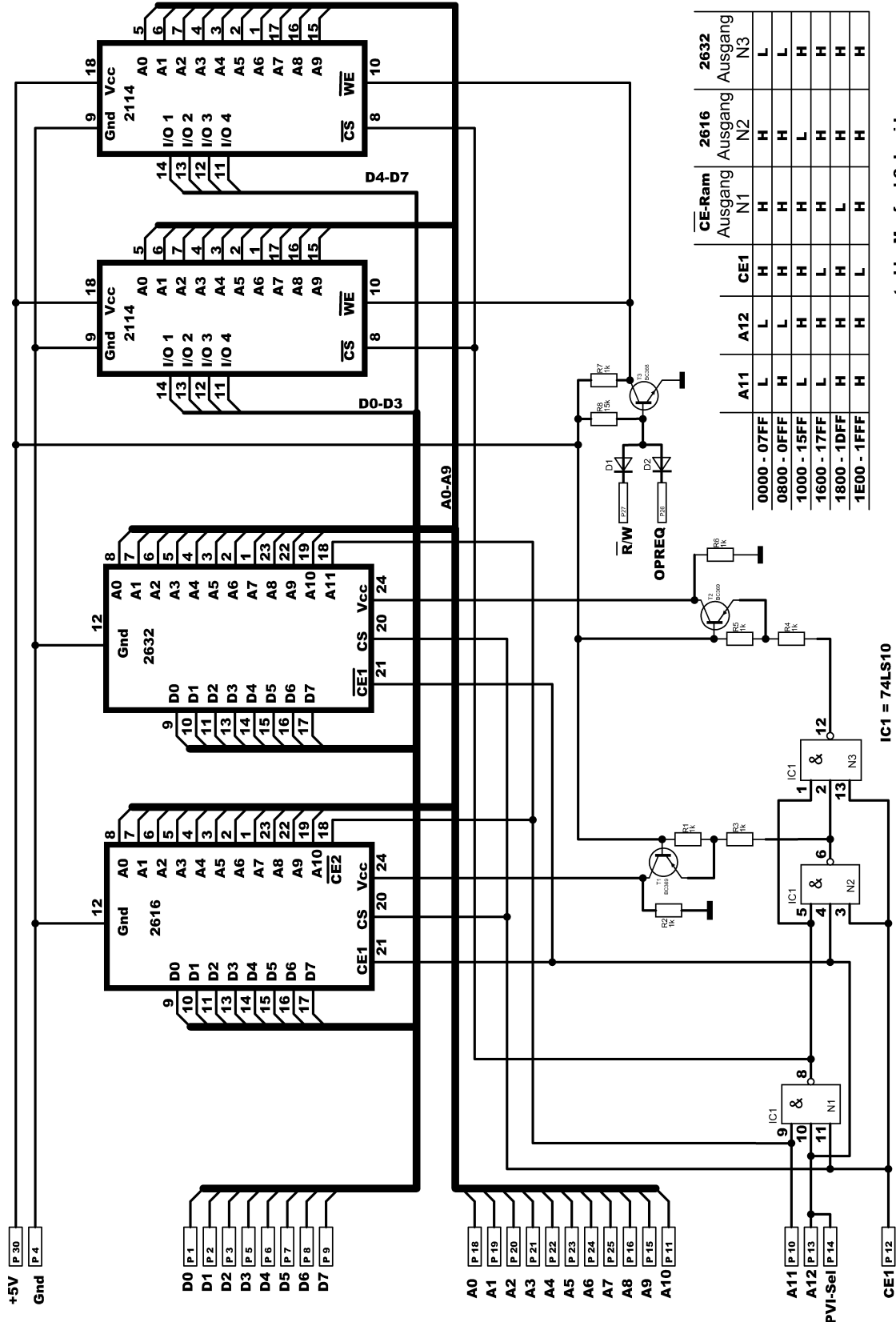
Es wäre demnach möglich, dass zumindest Teile des Programmcodes von „Schach 2“ davon abhängen, dass der RAM-Speicher über unterschiedliche Adressen erreichbar ist und unsere Schaltung deshalb zu dem beschriebenen Fehlverhalten führte. Denn ohne weiteres Zutun steuern die genannten Adressen 1800_{16} und $1C00_{16}$ tatsächlich zwei unterschiedliche Speicherzellen auf unserem 8 KiB großen SRAM-IC an. Wenn nun an einer Stelle des Spielcodes zunächst ein Wert an Adresse 1800_{16} gespeichert und an einer anderen Stelle explizit versucht würde, eben diesen Wert von Adresse $1C00_{16}$ zu lesen, so gelänge dies nur auf dem Originalmodul – nicht jedoch auf unserem Prototypen, wo unter der Adresse $1C00_{16}$ ein anderer als der zuvor gespeicherte Wert aufgefunden würde.

Um diese Hypothese zu überprüfen, wäre es einerseits möglich gewesen, den Programmcode von „Schach 2“ zu disassemblieren und ihn nach expliziten Adressierungen beider Speicherbereiche zu durchsuchen. Da dies zu zeitaufwändig erschien, experimentierten wurde stattdessen mit einer zusätzlichen Logikschaltung experimentiert, um die Spiegelung des Adressbereichs auf dem Spielmodul auch in unserer Hardware zu emulieren. Es stellte sich

⁵⁹ 1536 Adressen decken einen Speicherbereich von 1,5 KiB ab: 1024 Byte + 512 Byte.

⁶⁰ Vgl. auch den Abschnitt „Memory Maps“ in Jacobs, *Interton VC 4000 Coding Guide*.

Cassette 22 - Schach 2



created by Manfred Schneider

Abbildung 12: Schaltplan des „Schach-2“-Spielmoduls.

1.	1800	1 1 000 0000 0000	Beginn des SRAM-Speichers
⋮	⋮	⋮	
1024.	1BFF	1 1 011 1111 1111	Letzte Adresse im 1 KiB-Bereich
1025.	1C00	1 1 100 0000 0000	Beginn der Spiegelung
⋮	⋮	⋮	
1536.	1DFF	1 1 101 1111 1111	
1537.	1E00	1 1 110 0000 0000	PVI deaktiviert CE1 und damit die SRAM-ICs

Abbildung 13: Spiegelung des SRAM-Speichers auf dem Spielmodul „Schach 2“. Zur Verdeutlichung ist die den SRAM-ICs *fehlende* 11. Adressleitung (A10) in der Bitmuster-Spalte fett gedruckt.

jedoch heraus, dass der dafür nötige zusätzliche Platz auf der Platine sehr knapp würde. Als eine Minimalversion dieser Zusatzschaltung in einem Testaufbau obendrein dazu führte, dass zuvor funktionierende Spiele nun ebenfalls Fehler zeigten, wurden die Modifikationen wieder entfernt. Seltsamerweise funktionierte das Spiel bei einem erneuten Test nun fehlerfrei.

Im Nachhinein ist es schwierig zu klären, weshalb das Spiel zunächst nicht funktionierte und plötzlich doch fehlerfrei spielbar war. Eine erste Vermutung ging davon aus, dass eine fehlerhafte ROM-Datei die Ursache war. Tatsächlich zeigte ein Direktvergleich zweier aus unterschiedlichen Internetquellen stammenden ROM-Dateien des Spiels Unterschiede auf: in einer Datei fehlten die letzten 512 Byte. Letztendlich blieb diese Spur jedoch kalt, da der fehlende Speicherbereich in der einen Datei einer weiteren Schaltungseigenart der VC 4000 geschuldet ist. Wie die Wahrheitstabelle in Abbildung 12 zeigt, ist das CE1-Signal nicht nur im Adressbereich von $1E00_{16}$ aufwärts deaktiviert, sondern ebenso im Bereich von 1600_{16} – $17FF_{16}$. In diesen Adressraum, der direkt an den Spiel-ROM-Adressbereich anschließt und genau den fehlenden 512 Byte entspricht, wird der Speicherbereich der PVI-Register und einiger externer I/O-Schaltungen gespiegelt.⁶¹ Das heißt, Adressanfragen von der CPU in diesen Bereich gelangen ohnehin nicht in das SRAM auf dem Multi-ROM-Modul, sondern werden auf die entsprechenden PVI-Register oder den I/O-Bereich umgeleitet. Während des Auslesevorgangs, bei dem die vollständige, 6 KiB große ROM-Datei erzeugt wurde, wurden im Bereich über 1600_{16} lediglich die gerade aktuellen Werte der PVI-Register und der I/O-Hardware mit ausgelesen. Ein weiterer Test beider ROM-Dateien zeigte, dass sie beide funk-

⁶¹ Beispielsweise befindet sich an Adresse $1E80_{16}$ – und dank der Spiegelung auch an Adresse 1680_{16} – das sogenannte „NOISE“-Register, das eine externe analoge Verstärkerschaltung zur Lautstärkeregelung des PVI-Soundgenerators sowie einen diskret aufgebauten analogen Rauschgenerator für Explosionsgeräusche programmierbar macht, vgl. Schneider, *Schaltpläne der VC4000*, Datei *vc4000-Soundadapter.pdf*. An den darüberliegenden Adressen können die Signale des analogen Joysticks sowie die Taster auf den Spielcontrollern sowie der Konsolenfrontseite ausgelesen werden, vgl. Schneider, *Schaltpläne der VC4000*, Datei *vc4000-I-O.pdf*; Jacobs, *Interton VC 4000 Coding Guide*.

tionierten. Die vorübergehenden Probleme lagen also nicht an einer fehlerhaften ROM-Datei.

Ein weiterer Grund könnten minimale Unterschiede zwischen den zwei Prototypen des Multi-ROM-Moduls gewesen sein. Auf der ersten geätzten Platine lag das CE1-Signal des PVI nicht nur am Datenbustransceiver, sondern ebenso noch am entsprechenden *Chip-Select*-Eingang des SRAMs an. Eventuell könnte es hier zeitkritische Umschaltprobleme gegeben haben, die zu den Fehlern in „Schach 2“ führten. Auch waren hier nicht alle integrierten Schaltungen mit einem sogenannten Bypass-Kondensator zur Verminderung von störenden Schwankungen der Betriebsspannung versehen.⁶²

Dieser letzte Punkt könnte erklären, warum „Schach 2“ auch auf der zweiten Prototyp-Platine zunächst Fehler zeigte. Denn aus Mangel an ausreichenden 100 nF-Kondensatoren zum Zeitpunkt der Platinenbestückung waren diese zunächst gänzlich weggelassen worden. Erst während der Experimente mit der Logikschaltung für die RAM-Spiegelung wurden alle Bypass-Kondensatoren eingebaut. Da diese Testschaltung jedoch – wie oben beschrieben – fehlerhaft war, konnte sich der womöglich entscheidend positive Effekt der Bypass-Kondensatoren erst zeigen, nachdem die Modifikationen wieder entfernt wurden.

4 Das Ergebnis

Nachdem nun alle Original-Spiele auf unserem zweiten Prototypen erfolgreich zum Laufen gebracht wurden, ist das Projektziel des Entwurfs und der Konstruktion eines funktionierenden Multi-ROM-Moduls für die *VC 4000* erreicht. Nach ein paar letzten Anpassungen an der Arduino-Firmware kann das zu ladende Spiel nun auch direkt auf dem Multi-ROM-Modul ausgewählt werden. Hierzu wurde die vorübergehende Lösung mit der Textdatei zur Auswahl eines Spiel-ROMs auf der SD-Karte (siehe Abschnitt 3.6) zugunsten eines intuitiveren Systems aufgegeben. Der Nutzer wählt nun mit den Tastern auf der Platine eine zweistellige Nummer aus, die auf den Siebensegmentanzeigen dargestellt wird. Sobald die Auswahl Taste betätigt wird, versucht der Arduino eine ROM-Datei mit dieser zweistelligen Nummer als Dateinamen von der Speicherkarte zu laden. Abbildung 14 zeigt den aktuellen Aufbau des Multi-ROM-Moduls auf einer dritten Prototyp-Platine. Der Schaltplan in Abbildung 4 und das Platinenlayout in Abbildung 16 entsprechen diesem Aufbau. Hier sind die Siebensegmentanzeigen direkt auf der Platine angebracht, ohne – wie noch beim zweiten Prototypen – auf ein weiteres Breakout-Board angewiesen zu sein. Die LED-Segmente werden über zwei

⁶² Eine zusammenfassende Erklärung der elektronischen Funktion von Bypass-Kondensatoren liefert z. B. Kevin Ross. „Basic Circuits. Bypass Capacitors“. In: *Encoder. The Newsletter of the Seattle Robotics Society* (Juni 1997). URL: <http://www.seattlerobotics.org/encoder/jun97/basics.html> (besucht am 24.09.2015).

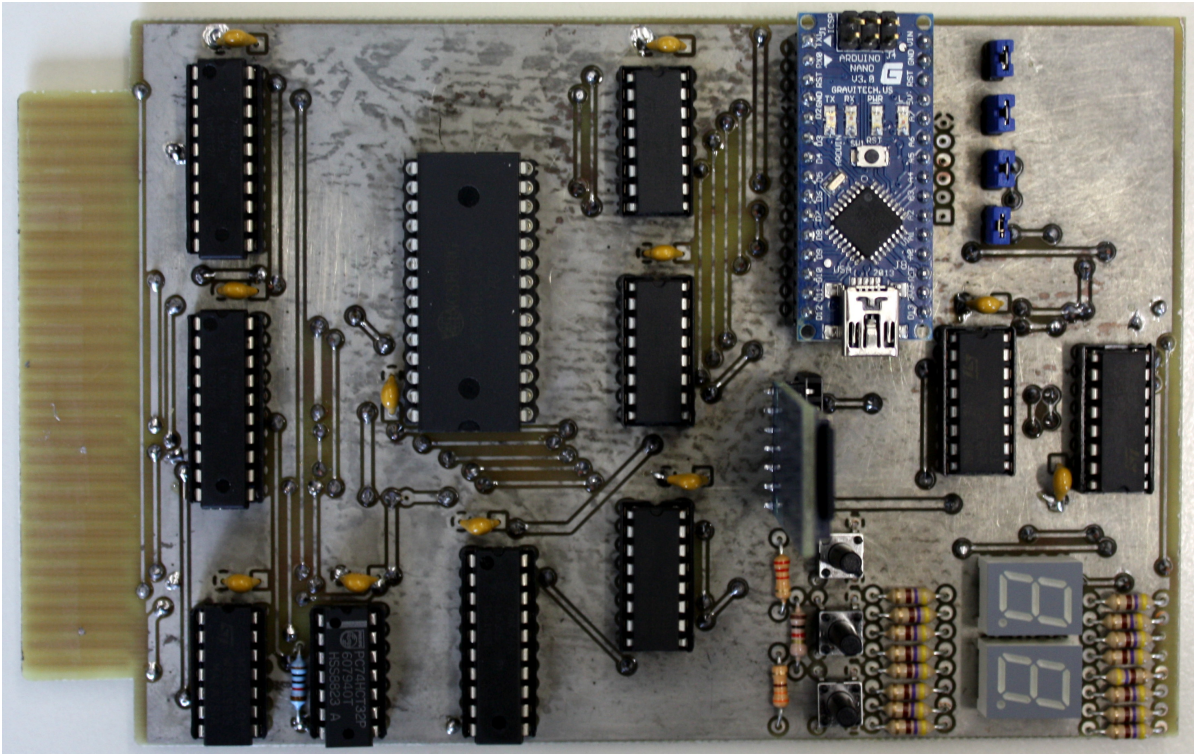


Abbildung 14: Ansicht der aktuellen Version (dritter Prototyp).

weitere Schieberegister vom Typ *74HC595* angesteuert.

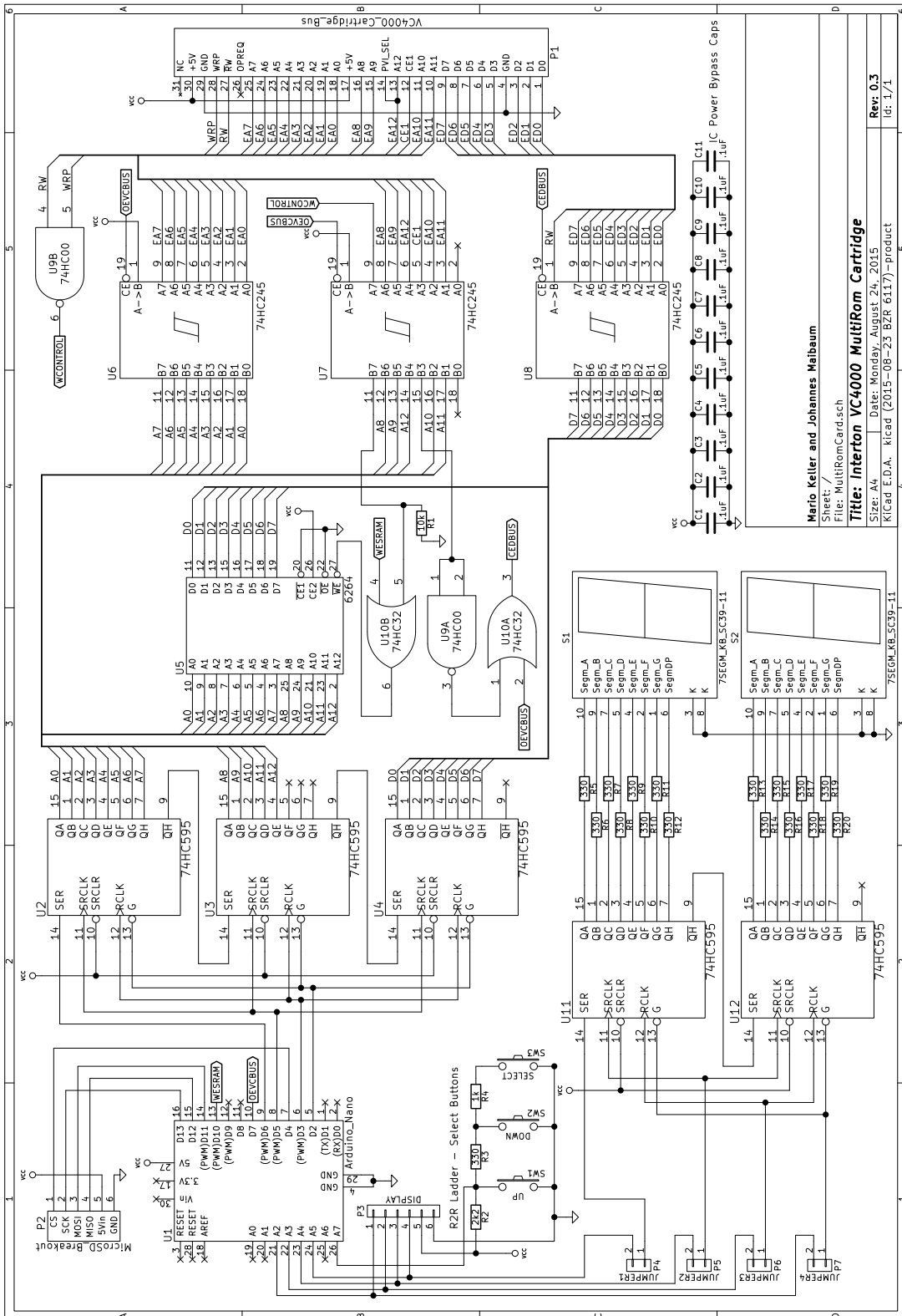


Abbildung 15: Schaltplan des entwickelten Multi-ROM-Moduls.

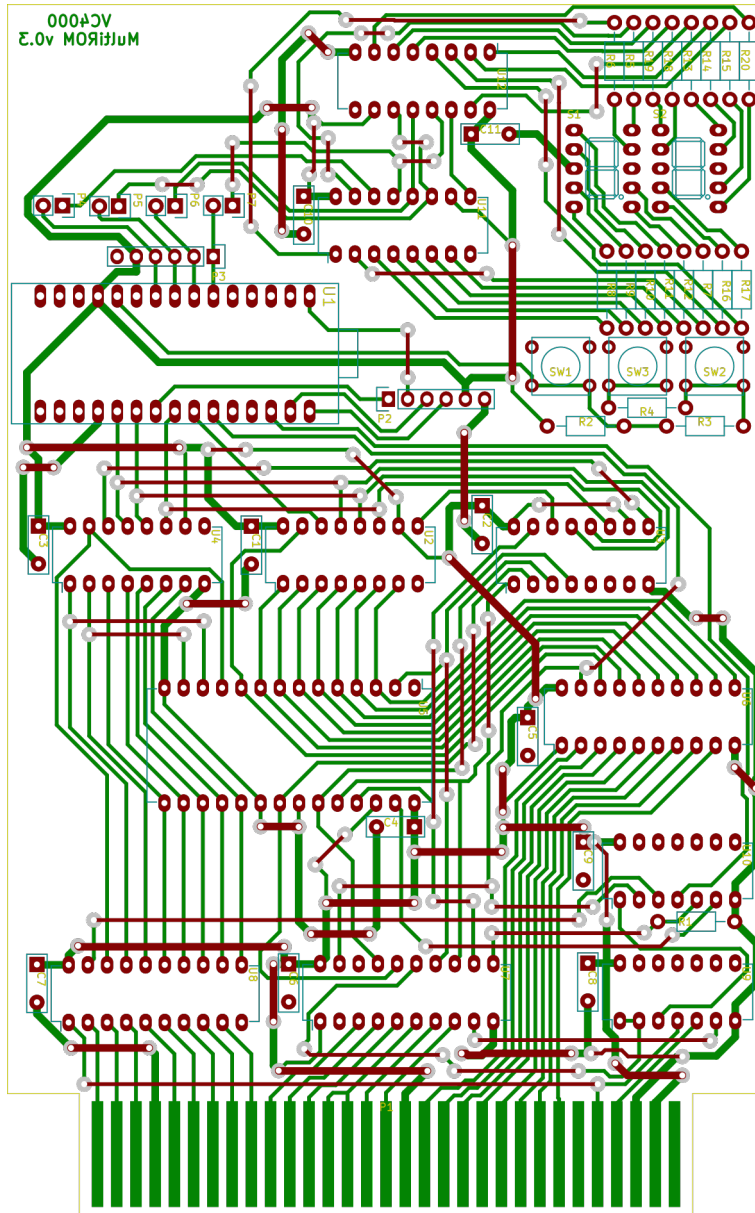


Abbildung 16: Grafischer Export des Platinenlayouts aus KiCad im Maßstab 1:1. Die Masseflächen wurden zur besseren Übersicht ausgeblendet.

4.1 Arduino-Firmwarecode

Der hier abgedruckte Firmwarecode für den Arduino entspricht der dritten Prototyp-Platine. Alle Entwicklungsschritte an diesem Code, am Schaltplan und am Platinenlayout können im Github-Repository des Projekts nachvollzogen werden.⁶³

```
1 #include <SD.h>
2 #include <SPI.h>           // Needed for SD lib
3
4 // Defines for Arduino port numbers
5 #define CLOCK             5 // SRCLK (A/D-Bus Shift Registers)
6 #define LATCH             3 // RCLK  (A/D-Bus Shift Registers)
7 #define DATA             6 // SER   (A/D-Bus Shift Registers)
8 #define OESHIFT          2 // _G   (A/D-Bus Shift Registers)
9 #define WESRAM           10
10 #define OEVCBUS          7 // OE for Bus Transceivers
11 #define SD_CS            4 // Chip Select for SD card
12
13 #define DISP_OE           A2 // 7-segment display
14 #define DISP_LATCH       A3 // 7-segment display
15 #define DISP_CLK         A4 // 7-segment display
16 #define DISP_DATA        A5 // 7-segment display
17
18 // Miscellaneous other defines
19 #define NOKEY             0
20 #define UPKEY             1
21 #define DOWNKEY          2
22 #define SELECTKEY        3
23 #define MAX_ROMS         99
24 #define MAX_ROM_SIZE     6144
25 #define GO_MSG           9 // 7-segment Message offset
26 #define ER_MSG           11 // 7-segment Message offset
27
28 // Global variables
29 byte lastkey = NOKEY;
30 byte message = GO_MSG;
31 byte romNumber = 1;
32 char romName[] = { 0, 0, 0, 0, 0, 0, 0 };
33 File romFile;
34 byte SevenSegChar[] = {
35     0x3f, // 0
```

⁶³ Keller und Maibaum, *VC4000 MultiRom Modul*. Der abgedruckte Code entspricht Commit 4d7ce272c9 vom 24. September 2015, Datei MultiRom/firmware/MultiRomVC4000/MultiRomVC4000.ino.

```

36 0x06 , // 1
37 0x5b , // 2
38 0x4f , // 3
39 0x66 , // 4
40 0x6d , // 5
41 0x7d , // 6
42 0x07 , // 7
43 0x7f , // 8
44 0x6f , // 9
45 0x63 , // o \
46 0x79 , // E |- Additional characters for status messages
47 0x50 // r /
48 };
49
50 void setup() {
51  pinMode(SD_CS, OUTPUT);
52  pinMode(CLOCK, OUTPUT);
53  pinMode(LATCH, OUTPUT);
54  pinMode(DATA, OUTPUT);
55  pinMode(WESRAM, OUTPUT);
56  pinMode(OESHIFT, OUTPUT);
57  pinMode(OEVCBUS, OUTPUT);
58  pinMode(DISP_LATCH, OUTPUT);
59  pinMode(DISP_CLK, OUTPUT);
60  pinMode(DISP_DATA, OUTPUT);
61  pinMode(DISP_OE, OUTPUT);
62
63  // Isolate MultiROM from console bus
64  disableVCBus();
65
66  // Enable 7-segment display
67  digitalWrite(DISP_OE, LOW);
68
69  // Initialize SD card
70  if (SD.begin(SD_CS)) {
71  // Wait for user selecting a ROM number
72  dispShowNumber(romNumber);
73  byte key = getKey(A7);
74  do {
75  if (key == UPKEY && lastkey != UPKEY && romNumber < MAX_ROMS)
76  ++romNumber;
77  else if (key == DOWNKEY && lastkey != DOWNKEY && romNumber > 1)
78  --romNumber;
79  dispShowNumber(romNumber);

```

```

80     lastkey = key;
81     key = getKey(A7);
82 } while (key != SELECTKEY);
83
84 // Construct ROM file name
85 if (romNumber < 10) {
86     romName[0] = '0';
87     itoa(romNumber, romName+1, 10);
88 } else {
89     itoa(romNumber, romName, 10);
90 }
91 strcpy(romName+2, ".bin"); // Append filename extension
92
93 // Open ROM file
94 romFile = SD.open(romName);
95
96 // Write ROM file to SRAM
97 if (romFile) {
98     unsigned romSize = static_cast<unsigned>(romFile.size());
99     if (romSize <= MAX_ROM_SIZE) {
100         for (unsigned i = 0; i < romSize; i++) {
101             write2RAM(i, romFile.read());
102         }
103     } else {
104         // ROM size too big.
105         message = ER_MSG;
106     }
107     romFile.close();
108     enableVCBus();
109 } else {
110     // ROM file could not be opened.
111     message = ER_MSG;
112 }
113 } else {
114     // SD card not found or otherwise failing initialization
115     message = ER_MSG;
116 }
117 }
118
119 void loop() {
120     // Status indication
121     dispMessage(message);
122 }
123

```

```

124 void disableVCBus() {
125     digitalWrite(WESRAM, HIGH);
126     digitalWrite(OESHIFT, LOW);
127     digitalWrite(OEVCBUS, HIGH);
128 }
129
130 void enableVCBus() {
131     digitalWrite(WESRAM, HIGH);
132     digitalWrite(OESHIFT, HIGH);
133     digitalWrite(OEVCBUS, LOW);
134     delay(100);
135     // handover R/W control to VC4000
136     digitalWrite(WESRAM, LOW);
137 }
138
139 void write2RAM(unsigned int address, byte data) {
140     digitalWrite(OESHIFT, LOW);
141     digitalWrite(WESRAM, HIGH);
142     shiftOut(address, data);
143     digitalWrite(WESRAM, LOW);
144     digitalWrite(WESRAM, HIGH);
145 }
146
147 void shiftout(unsigned int address, byte data) {
148     byte lowbyte = address & 0xff;
149     byte highbyte = address >> 8;
150
151     digitalWrite(LATCH, LOW);
152     shiftOut(DATA, CLOCK, MSBFIRST, data);
153     shiftOut(DATA, CLOCK, MSBFIRST, highbyte);
154     shiftOut(DATA, CLOCK, MSBFIRST, lowbyte);
155     digitalWrite(LATCH, HIGH);
156     digitalWrite(LATCH, LOW);
157 }
158
159 byte getKey(int pin) {
160     int v = analogRead(pin);
161     delay(50);
162     v = analogRead(pin);
163     if (v < 600 && v > 300)
164         return DOWNKEY;
165     if (v < 300 && v > 100)
166         return SELECTKEY;
167     if (v < 100)

```

```

168     return UPKEY;
169     return NOKEY;
170 }
171
172 void dispShowNumber(byte num) {
173     byte i = num % 10;
174     byte j = (num - i) / 10;
175     digitalWrite(DISP_LATCH, LOW);
176     shiftOut(DISP_DATA, DISP_CLK, MSBFIRST, SevenSegChar[i]); // digitTwo
177     shiftOut(DISP_DATA, DISP_CLK, MSBFIRST, SevenSegChar[j]); // digitOne
178     digitalWrite(DISP_LATCH, HIGH);
179 }
180
181 void dispMessage(byte offset) {
182     digitalWrite(DISP_LATCH, LOW);
183     shiftOut(DISP_DATA, DISP_CLK, MSBFIRST, SevenSegChar[offset + 1]);
184     shiftOut(DISP_DATA, DISP_CLK, MSBFIRST, SevenSegChar[offset]);
185     digitalWrite(DISP_LATCH, HIGH);
186 }

```

5 Retrocomputing als medienarchäologische Methode

Die Situierung des praktisch durchgeführten medientechnischen Projekts im Masterstudienang Medienwissenschaft verlangt nach einer entsprechenden *medientheoretischen* Rahmung des zuvor Gesagten, das sich einer technischen Beschreibung entsprechend hauptsächlich an den technisch notwendigen Einzelschritten und dem dafür zweckdienlichen praktischen Wissen entlanghangeln musste. Aufgrund des materiellen Untersuchungsgegenstands der Videospielekonsole *Interton VC 4000* stammt der Großteil dieses Wissens aus der Elektrotechnik oder genauer: ihrem Teilgebiet der Digitalelektronik. Die zum Erreichen des Projektziels verwendete Mikrocontrollerplattform *Arduino* bezieht auch Werkzeuge und Praktiken aus der zeitgenössischen praktischen Informatik mit ein: Die Programmierung innerhalb von aktuellen Hochsprachen-Entwicklungsumgebungen, die ihrerseits das Einbinden von fertigen Programmbibliotheken umfasst. Im Folgenden soll dieses praktische Projekt kurz medientheoretisch reflektiert werden.

Von der Kulturwissenschaft würde die Entwicklung einer Flash-Speicher-Schnittstelle, um Daten von modernen Flash-Speichern wie auf ältere Videospielekonsolen laden zu können, wohl als typische Aktivität der kulturellen Praxis namens *Retrocomputing* verstanden werden. Stark komprimiert definiert die Kulturwissenschaft Retrocomputing als „a set of diverse

practices involving contemporary engagement with old computer systems. Such practices are primarily private and non-professional, [...].“⁶⁴ Wenn man weiter den Versuch wagen möchte, verschiedene kulturwissenschaftliche Arbeiten zum Retrocomputing ähnlich knapp zusammenzufassen, kann darin die Tendenz ausgemacht werden, den Antrieb für Retrocomputing vornehmlich in einer „nostalgischen Sehnsucht“ der sie betreibenden Personen zu suchen.⁶⁵ Die Schlussfolgerungen kulturwissenschaftlicher Arbeiten zum Retrocomputing sind dennoch vielfältiger Art: Zu nennen ist erstens Takhteyevs und DuPonts pragmatische Forderung an kulturelle Gedächtnisorganisationen wie Technikmuseen, „to appreciate the technical knowledge possessed by retrocomputing participants and the work they have done“, also mit Hilfe des praktischen Wissens der Retrocomputing-Kultur das drängende Problem der Erhaltung obsoleter Hard- und Software in einem funktionsfähigen Zustand endlich systematisch anzugehen.⁶⁶ Sebastian Felzmann wiederum analysiert in seinem Artikel den primär sozialen Wunsch der Retro gamer als *Rezipienten* der einschlägigen Videospieleästhetiken und -inhalte, über das persönliche Spielen und den Austausch darüber mit anderen Retro Spielern „eine Rückkehr zu Altbekanntem zu suchen, welches ihnen eine vertraute Identifikationsbasis für die Definition ihres eigenen Selbst innerhalb einer von Veränderungen geprägten Moderne liefert.“ Mit diesem Ansatz untersucht er das Interesse kommerzieller Hersteller von „Retro-Remakes“, die Nostalgie der Spieler in ökonomischen Profit umzuwandeln, indem sie erfolgreiche alte Spiele für aktuelle Spielhardware neu auflegen. Abschließend stellt Felzmann daher die Frage, „was ein Spiel nun zu einem Klassiker macht“.⁶⁷

Innerhalb der Medienarchäologie, wie sie auch an der Berliner Humboldt-Universität betrieben wird, wird Retrocomputing als praktische Methode jedoch vor allem epistemologisch weitergedacht.⁶⁸ Stefan Höltgen versteht Retrocomputing innerhalb seines laufenden Forschungsprojekts zur „Archäologie des frühen Mikrocomputers und seiner Programmierung“ als „*implizit vollzogene* medienarchäologische Praxis [...], die noch ihrer Explikation harret.“⁶⁹ Demnach können die Praktiken des Retrocomputings als Erweiterung des Methodenkanons

64 Yuri Takhteyev und Quinn DuPont. „Retrocomputing as preservation and remix“. In: *Library Hi Tech* 31.2 (2013), S. 355–370, hier S. 357.

65 Sebastian Felzmann. „Been there, done that. Mediennostalgie als kreative Praxis zur Schaffung neuer retroider Spiele“. In: *SHIFT – RESTORE – ESCAPE. Retrocomputing und Computerarchäologie*. Hrsg. von Stefan Höltgen. Winnenden: CSW, 2014, S. 25–39, hier S. 31. Der Beitrag zitiert viele einschlägige Arbeiten zum Retrocomputing und Retro gaming.

66 Takhteyev und DuPont, „Retrocomputing as preservation and remix“, S. 365.

67 Felzmann, „Been there, done that“, S. 37.

68 Eine Einführung über die internationalen Strömungen in der Medienarchäologie liefern Erkki Huhtamo und Jussi Parikka, Hrsg. *Media Archaeology. Approaches, Applications and Implications*. Berkeley, Los Angeles und London: University of California Press, 2011.

69 Stefan Höltgen. „OPEN_. Ein Vorwort zur Computerarchäologie“. In: *SHIFT – RESTORE – ESCAPE. Retrocomputing und Computerarchäologie*. Hrsg. von dems. Winnenden: CSW, 2014, S. 11–23, hier S. 13.

der Medienarchäologie verstanden werden, als Erweiterung einer vornehmlich textorientierten und theoretischen Disziplin um praktische und zuweilen pragmatische Vorgehensweisen.

Man kann behaupten, die Retrocomputing-Kultur käme – trotz oder dank ihrer Nostalgie – bereits durch einfaches *praktisches* Ausprobieren zum gleichen Ergebnis wie der Medientheoretiker Kittler in seinen „technischen Schriften“. Denn dass „eine Maschine [...] also zugleich weniger und mehr [kann], als ihre Datenblätter zugeben“, wird jeder Retrogamer bestätigen, der schon einmal mit einer unzureichenden Emulationssoftware konfrontiert war (wie auch die Teilnehmer des Assemblerworkshops aus dem dieses Projekt hervorging, siehe Abschnitt 1). Die Feststellung Kittlers, die in den 1980er Jahren noch eine Revolution der Literaturwissenschaft auslösen konnte, reicht der Retrocomputing-Gemeinschaft als Erklärung jedoch nicht aus. Technisches Scheitern spornt ihre Teilnehmer erst zu weiteren Anstrengungen an, den Ursachen der Diskrepanz zwischen Datenblättern, Hardware- und Softwareverhalten erst recht auf den Grund zu gehen und sie zu dokumentieren. Wie auch im vorliegenden Projekt regt dies nicht nur die Suche nach genaueren Datenblättern und (häufig von anderen Retrocomputing-Enthusiasten geschriebenen) technischen Beschreibungen an, also weiteren klassischen *Texten*. Retrocomputing lässt darüber hinaus die fragliche Technik selbst zu Wort kommen und untersucht ihr tatsächliches Verhalten. Das heißt, ihre Ein- und Ausgangssignale durch entsprechende Messgeräte gezielt auszumessen (siehe Abschnitt 3.4). Technische Medien betreiben auf diese Weise ihre eigene Archäologie, also eine *Medienarchäologie* im Wortsinne, wie eines der Kernargumente des Medienarchäologen Wolfgang Ernst lautet.⁷⁰ Da die Taktraten digitaler Computer die Fähigkeiten des menschlichen Wahrnehmungsapparats weit überschreiten, bleibt häufig keine andere Wahl, als sie mit anderen technischen Medien auszumessen.

Die aktive Praxis des Retrocomputings, die tatsächlich an, mit und vor allem *in* Computersystemen arbeitet, liefert hier ein wirksames Plädoyer für mehr Hands-on auch in einer traditionell vornehmlich geisteswissenschaftlichen Disziplin wie der Medienwissenschaft. Aktives Retrocomputing, das die Geräte tatsächlich *öffnet*, liefert auf diese Weise nicht nur politische Argumente gegen eine Vereinnahmung von (immer auch politisch-sozialen) Bewegungen wie Richard Stallmans „freier Software“⁷¹ durch den „freien Markt“,⁷² sondern auch

⁷⁰ Vgl. z. B. Wolfgang Ernst. „Media Archaeography. Method and Machine versus History and Narrative of Media“. In: *Media Archaeology. Approaches, Applications and Implications*. Hrsg. von Erkki Huhtamo und Jussi Parikka. Berkeley, Los Angeles und London: University of California Press, 2011, S. 239–255, hier S. 245.

⁷¹ Richard Stallman. *The GNU Manifesto*. Mehrfach überarbeitete Version auf [gnu.org](https://www.gnu.org/gnu/manifesto). (Erstveröffentlichung 1985). 12. Juni 2015. URL: <https://www.gnu.org/gnu/manifesto> (besucht am 24. 09. 2015).

⁷² Der Publizist Evgeny Morozov hat für die Vereinnahmung des Begriffs „open“ durch „innovative“ Technologiestartups den Begriff des „openwashings“ geprägt, siehe Evgeny Morozov. „Open and Closed“. In: *New York Times* (16. März 2013). URL: <http://www.nytimes.com/2013/03/17/opinion/sunday/morozov-open-and-closed.html> (besucht am 24. 09. 2015). Eine ausführlichere Diskussion seiner Argumente findet

handfeste epistemologische Argumente für eine technikaffine Medienwissenschaft.

5.1 Medientheoretische Erkenntnisse durch aktives Retrocomputing

So brachte dann auch das in diesem Projekt aktiv betriebene Retrocomputing einige medientheoretisch interessante Punkte ans Licht, die weitere, konkret *technologisch belegbare* Argumente zur vonseiten der Medientheorie geforderten „akademisch-präzise[n] Definition von Retrocomputing“ beitragen.⁷³ So zeigte sich beim Aufbau des Multi-ROM-Moduls für die *VC 4000*, dass Retrocomputing eine viel umfassendere kulturelle Praxis ist, als dass ihr Vorhandensein rein durch Nostalgie der Beteiligten erklärbar wäre.

Hierzu sei noch einmal auf den *SN74LCV125A* von Texas Instruments auf dem verwendeten MicroSD-Breakout-Board verwiesen, der den TTL-Pegel der Signale vom Arduino auf das modernere CMOS-Level von 3,3 V herabsenkt bzw. die Signale von der SD-Karte zur Prozessierung im Mikrocontroller entsprechend wieder auf 5 V anhebt. Solche integrierten Schaltkreise schlagen Brücken zwischen Generationen von Logikfamilien. Im Gegensatz zur Video-spielindustrie, die dem Endverbraucher als Nutzer alle paar Jahre *neue, schnellere, hochauflösendere, immersivere* Spielkonsolen liefert, woraufhin alles Spielen auf der damit „obsolet“ gewordenen Hard- und Software der Vorgängergeneration zum Retrocomputing degradiert wird – mit der Folge, dass sich, der Nostalgiedefinition nach, nun lediglich noch Enthusiasten für diese interessieren sollten –, werden von der Halbleiterindustrie ganze Schaltkreisfamilien einzig zu dem Zweck neu entwickelt, dass obsoletere integrierte Schaltungen weiterhin Einzug in Neuentwicklungen finden oder zumindest Schnittstellen zu modernen Systemen konstruiert werden können. In einer „Application Note“ der Fairchild Semiconductor Corporation aus dem Jahr 1984 liest man über die Existenzbegründung der 74HCT-Logikfamilie:

Why does HCT exist? Ideally, when a designer sits down to design a low power high speed system, he would like to use 54HC/74HC, and CMOS LSI components. Unfortunately, due to system requirements, he may have to use NMOS microprocessors or bipolar peripherals or bipolar logic (54S/74S, 54F/74F, 54ALS/74ALS, or 54AS/74AS) because either the specific function does not exist in CMOS or the CMOS device may not have adequate performance. Since the system designer still desires to use HC where possible, he will mix HC with these products. If these devices are specified to be TTL compatible, incompatibilities may result at the interface between the TTL, NMOS, etc. and HC.⁷⁴

sich z. B. in Evgeny Morozov. „The Meme Hustler. Tim O’Reilly’s crazy talk“. In: *The Baffler* 22 (2013), S. 66–75. URL: <http://thebaffler.com/articles/the-meme-hustler> (besucht am 24. 09. 2015). Zum Innovationsbegriff, siehe Evgeny Morozov. „Die Rede von der Innovation. Der Jargon des Silicon Valley fängt sie alle ein“. Aus dem Englischen übers. von Matthias Fienbork. In: *Frankfurter Allgemeine Zeitung* (18. Apr. 2014). URL: <http://www.faz.net/-hbj-7ofqj> (besucht am 24. 09. 2015).

73 Höltgen, „OPEN“, S. 12.

74 Fairchild Semiconductor Corporation, Hrsg. *An Introduction to and Comparison of 74HCT TTL Compatible*

Bei der hier angedeuteten möglichen Inkompatibilität zwischen älteren 74er TTL-Serien und dem modernen 74HC-Äquivalent dreht es sich ebenfalls um die Logikpegel der verschiedenen Familien. Wie der Leser der Application Note im nächsten Absatz erfährt, bedeutet das Prädikat „TTL-kompatibel“ einer beliebigen integrierten Digitalschaltung mitnichten, dass der hohe Signalpegel genau bei 5 V liegen muss, wie häufig kolportiert: „Notice that the output high level of a TTL specified device will not be guaranteed to be recognized as a valid logic high input level by HC. A TTL output will be equal to or greater than 2.4V, but an HC-MOS input needs at least 3.15V.“⁷⁵ In den meisten Fällen sei eine direkte Verbindung von TTL-kompatiblen Schaltkreisen mit Logikbausteinen aus der 74HC-Reihe (wie sie auch im vorliegenden Projekt zum Einsatz kamen) zwar unproblematisch, da die meisten älteren TTL-Bausteine diese Minimalanforderungen weit überböten. Für mögliche Randfälle wurde jedoch die Sonderfamilie 74HCT entwickelt, die einen hohen Eingangspegel TTL-kompatibel schon ab 2,4 V erkennt.

Die Halbleiterindustrie weiß also um die Tatsache, dass alte Hardware zwar technisch überholt sein kann, viele Entwickler elektronischer Schaltungen jedoch auf gewisse *bewährte* Bauteile zurückgreifen wollen und manchmal sogar müssen. Hans Camenzind schreibt rückblickend über den von ihm entwickelten 555-Timer-IC: „Looking at the design now, 33 years later, there are many areas where it can be improved with the design techniques we have learned since and with the enormous benefit of computer simulation.“ Aber er gibt – natürlich nicht ohne Stolz – zu: „Except for a CMOS version, the design has never been changed.“⁷⁶ Es wurde nie verändert, weil Milliarden von bewährten elektronischen Geräten, die sich im täglichen Einsatz befinden, auf die in seinem Datenblatt spezifizierte Funktionalität hin entworfen sind und von ihr abhängen.

Wenn man dies berücksichtigt, kann das Feld des Retrocomputings beträchtlich erweitert werden und es hängt nicht länger nur vor der Nostalgie von Enthusiasten und dessen ökonomischer Ausnutzung durch die Unterhaltungsindustrie ab. Stattdessen hängen ganze Technologiezweige davon ab. Der 555, Pegelumsetzer wie der *SN74LCV125A* oder die komplette 74HCT-Logikfamilie können demnach als aktives Retrocomputing der Halbleiterindustrie verstanden werden. Wenn man Retrocomputing anders definieren wollte, könnte man es auch als aktiven und daher notwendig invasiven Versuch verstehen, *bewährte* Systeme so lange als möglich im produktiven Einsatz zu halten. Nostalgie wäre aus dieser Perspektive dann möglicherweise ein Resultat dieses Vorhabens, jedoch nicht länger seine Ursache.

CMOS Logic. Application Note 368. März 1984. URL: <https://www.fairchildsemi.com/application-notes/AN/AN-368.pdf> (besucht am 24.09.2015), S. 1.

⁷⁵ Fairchild Semiconductor Corporation, *Introduction to 74HCT*, S. 1.

⁷⁶ Camenzind, *Designing Analog Chips*, S. 11-3.

Seit Kittler weiß die Medientheorie, dass auch Datenblätter und Application Notes zu konsultieren sind, wenn der Untersuchungsgegenstand ein technisches Medium ist. Eine durch Retrocomputing auch praktisch instruierte und informierte Medienarchäologie erinnert sich jedoch auch an Kittlers Warnung vor unzureichenden technischen Dokumentationen. Ihr Lesen allein entbirgt demnach längst nicht alle möglichen epistemologischen Erkenntnisse. Durch die aktive technische Beschäftigung mit den in Datenblättern beschriebenen Geräten, Bauteilen, ihren Signalen im laufenden Betrieb und ihrer Software gelangt Retrocomputing hier weiter.

5.2 Medienarchäologie versus Mediengeschichte?

Die Medienwissenschaft sollte diese zusätzliche Erkenntnisdimension nicht von vorne herein ausschließen, wie es einige Teilnehmer der seit einiger Zeit innerhalb der Disziplin schwellenden Debatte um Kittlers Medientheorie tun. So versucht zum Beispiel Claus Pias, Kittlers laienhaftes Löten und Programmieren in den 1980er und 90er Jahren mit dem Verweis auf die sich vermeintlich rapide ändernde Technologie zu historisieren:

Beides – das Konzept der Zweckentfremdung und des praktizierenden Medienverstehens, die Verallgemeinerung des Einzelplatzrechners und die These vom „medientheoretischen Apriori“ – ist nicht nur zeittypisch und an einen bestimmten historischen Stand von Hard- und Software gebunden, sondern auch in der Geschichte und der Geschichtsschreibung der Computerkultur bereits vorgeprägt. Deswegen funktionieren einige der theoretischen Konsequenzen nicht mehr, sobald sich diese Technik ändert – was sie rapide getan hat. [...]

Zahllose vernetzte digitale Geräte bevölkern unseren Alltag und kommunizieren (oft ohne Umweg über Subjekte) untereinander. Solche Phänomene lassen sich nur schwerlich beschreiben, indem man einen Schreibtisch-PC in Assembler oder C programmieren lernt oder Grafikbibliotheken optimiert – so sinnvoll das gewesen sein mag, um grundlegende Arbeitsweisen des Computers zu begreifen und ein Denken in Notationssystemen und Algorithmen zu üben.⁷⁷

Etwas polemisch lässt sich auf dieses Plädoyer gegen mehr Hands-on zunächst mit dem direkten Hinweis auf die Retrocomputing-Gemeinschaft antworten, die Webbrowser für obsoletere Hardware längst entwickelt hat, wie z. B. den *Singular Browser* für den Commodore C64.⁷⁸ Will man etwas differenzierter vorgehen, lässt sich an Pias' Argument zugute halten, dass eine Medienwissenschaft, die ihres ebenso geisteswissenschaftlichen Erbes aus Literatur- und Geschichtswissenschaft nicht Unrecht tun möchte, die Antworten auf ihre Fragen

⁷⁷ Pias, „Friedrich Kittler und der ‚Mißbrauch von Heeresgerät‘“, S. 33 f.

⁷⁸ Vgl. Singular, Soci und The Dreams (Pseudonyme). *Singular browser*. URL: <http://singularcrew.hu/browser> (besucht am 24.09.2015). Passenderweise ist der Quellcode dafür ausgerechnet in 6502-Assembler und in C geschrieben.

ebensowenig ausschließlich in Quellcodes und Schaltplänen findet, sondern eben auch in klassischen Texten.

Interessanterweise lassen sich in der Technikgeschichte sowohl unterstützende Aussagen, wie auch Gegenargumente zu Pias finden. Der Gewährsmann, der beide Aspekte vereint, ist der Technikhistoriker Michael Mahoney. Sein Grundansatz ist, die Geschichtsschreibungen zum Digitalcomputer zunächst von der eigentlichen Maschine wegzulenken, was er als „Dezentrierung der Maschine“ bezeichnet,⁷⁹ indem er auf die unterschiedlichen disziplinären Hintergründe und Geschichten derjenigen Gruppen verweist, welche die ersten elektronischen Digitalcomputer bauten: „The computer has [...] little or no history of its own. Rather, it has *histories* derived from the histories of the groups of practitioners who saw in it [...] the potential to realise their agendas and aspirations.“⁸⁰ Der Fokus auf die unterschiedlichen Kontexte, in denen und für die Computer gebaut wurden, dient Mahoney dazu, darauf hinzuweisen, dass z. B. die mathematischen *number crunchers*, wie sie von John von Neumann, Herman Goldstine und anderen gebaut wurden, *andere* Computer waren als die Datenverarbeitungsmaschinen von IBM, die in Wirtschaftsunternehmen eingesetzt wurden – obwohl alle dem abstrakten Funktionsprinzip der Turingmaschine folgten. Dabei modifizierten diese unterschiedlichen Gruppen sowohl „den Computer“ und pluralisierten damit „seine Geschichte“, genauso wie die gebauten Computer die sie konstruierenden Gruppen und ihre Disziplinen veränderten:

Different groups of people saw different possibilities in computing, and they had different experiences as they sought to realise those possibilities. One may speak of them as “communities of computing”, or perhaps as communities of practitioners that took up the computer, adapting to it while they adapted it to their purposes.⁸¹

Dies ist auch Pias' Kernargument: Kittlers Hacken in Assembler und C geschah auf einem historisch anderen Computer als dem, der zum Verfassen dieser Arbeit genutzt wurde. Auch die Medienwissenschaft hat, initiiert durch Kittlers Löten und Programmieren, den Computergeschichten ihre eigene Variante hinzugefügt. Der entsprechend selbst konstruierte Computer wirkt auf die Disziplin zurück. Kittler war kein jugendlicher Hacker mehr, als er anfang zu programmieren, sondern „ein brillianter, habilitierender Germanist.“⁸² Im Sinne Mahoneys und Pias' ließe sich demnach formulieren: Durch Kittlers Computer und den daran und damit arbeitenden Germanisten wandelte sich ein Zweig der Germanistik zur Medienwissenschaft. Was es für Kittlers Theorie bedeutet, wenn die „These des *medientechnischen Apriori*“ rekursiv

79 Mahoney, „The histories of computing(s)“, S. 121, Abschnitt „Decentering the Machine“.

80 Ebd., S. 119.

81 Ebd., S. 124.

82 Pias, „Friedrich Kittler und der ‚Mißbrauch von Heeresgerät‘“, S. 39.

auf sich selbst angewandt wird, ist auch eine Fragestellung der Medienwissenschaft.⁸³

Aber auch der Technikhistoriker Mahoney bezweckt mit seiner „Dezentrierung der Maschine“ keineswegs eine vollständige *Abkehr* vom technischen Objekt und seinen Funktionsweisen hin zu rein klassisch historischem Forschen an den Texten der Computernutzergruppen. Er betont ebenso: „Technology is not a literate enterprise, but a visual, tactile one. Its practitioners think not with words but, as Derek de Solla Peirce so deftly phrased it, ‘with their fingertips’.“⁸⁴ Und genauso sollten sich auch Computerhistoriker verhalten. Der Blick auf das, was verschiedene Gruppen von Praktikern mit und an Computern tatsächlich getan haben und tun, entzaubere nämlich auch die Mythen vom ewigen, alles revolutionierenden technischen Fortschritt:

With some recent exceptions, the history of computing has been centered on the machine, tracing its origins back to the abacus and the first mechanical calculators and then following its evolution through the generations of mainframe, mini, and micro. [...] Once invented, the computer evolves naturally into the PC as its present most visible form,⁸⁵ rather than into a variety of coexisting, mutually supportive forms (as if mainframes disappeared with the invention of the minicomputer). Its progress is inevitable and unstoppable, its effects revolutionary.

Chronicling the revolution, that machine-centered history reinforces the hype and with it what one might call the “impact theory” of the relation of technology and society. There is society strolling along, minding its own business, and, wham!, it gets impacted and is left reeling by a revolutionary technology, which changes everything overnight or in some similar short time. [...] From that perspective, society breaks up into two classes – those who are on the train and those who are not – and the latter are hopeless [...]. In our field, you’re either a digeratus or a dinosaur.⁸⁶

Bei der Beantwortung der Frage, ob Kittlers Theorie des aktiven „Mißbrauchs“ von technischen Geräten wie im Retrocomputing also aufgrund des technischen Fortschritts vom raumfüllenden, aber isolierten Mainframe zu den weltweit vernetzten Smartphones an allen Ohren mehr als eine historische Notiz wert ist, lohnt demnach sowohl der Blick *in die Geräte*, der zeigt, dass auch Smartphones immer noch Implementierungen der gleichen abstrakten Turingmaschine sind, wie auch der Blick *auf die Praxis des Retrocomputings*. Dank dieser Praxis hat dieses Projekt bewiesen, dass auch Spielkonsolen aus den späten 1970er Jahren von modernen Flash-Speichern lesen können. Ebenso gelingt es heute, auf Heimcomputern der 1980er Jahre Nachrichten in sozialen Netzwerken wie Twitter zu schreiben und zu lesen.⁸⁷

83 Pias, „Friedrich Kittler und der ‚Mißbrauch von Heeresgerät‘“, S. 33.

84 Mahoney, „The histories of computing(s)“, S. 130.

85 Im Jahr 2015, zehn Jahre nachdem Mahoney diese Zeilen schrieb, müssten hier wohl die von Pias erwähnten „zahllosen vernetzten digitalen Geräte“, also hauptsächlich Smartphones stehen.

86 Mahoney, „The histories of computing(s)“, S. 121.

87 Vgl. Christian Klaß. „Twittern mit dem C64“. In: *Golem.de* (15. Juni 2009). URL: <http://www.golem.de/0906/67762.html> (besucht am 24. 09. 2015).

5.3 Senden und Empfangen

Natürlich muss das größtenteils informelle, praktische Wissen, wie es im Retrocomputing an der Tagesordnung ist, historisch kritisch hinterfragt werden. Allzu oft lesen sich die von Retrocomputing-Enthusiasten verfassten Technik-, Geräte- und Firmengeschichten wie Legenden und Mythen. Belastbare historische Quellenangaben sucht man nicht selten vergeblich.⁸⁸ Als ein kurzes Beispiel für eine medienhistorisch kritische Recherche soll hier nochmals auf den Begriff des *Transceivers* verwiesen werden, der im Zuge dieses Projekts in Form des 74HC245 Bus Transceiver als „Sender-Empfänger“ für die technisch korrekte Orchestrierung der Signale auf den Adress- und Datenbussen nötig wurde (siehe Abschnitt 3.3).

Laut der deutschen Wikipedia, deren Autoren sich für die Angabe, dass „das Kofferwort [...] in den 1930er Jahren [entstand]“,⁸⁹ ihrerseits auf das Oxford English Dictionary beziehen,⁹⁰ scheint das Wort aus der Radio- und Funktechnik zu stammen: „Damals war bei Funkgeräten noch ungewöhnlich, weder ihr Funkmodul zwischen Senden und Empfangen umzuschalten, noch drei Antennen ausrichten zu müssen, sondern einfach beides gleichzeitig zu können.“⁹¹ Eine kritische Recherche in den online verfügbaren Patenten aus der fraglichen Zeit kommt jedoch zu einem anderen Schluss.

Während in Patenten zur Radio- und Funktechnik ausschließlich von einem „Combined Wireless Sending and Receiving System“ geschrieben wird,⁹² taucht der Begriff des *transceivers* erstmals in einem Patent von Harvey Hayes auf, das am 22. Oktober 1932 beantragt wurde: „It is to be noted that this apparatus serves equally well as a receiver or transmitter of sound, to which type of device I have heretofore applied the name of transceiver.“⁹³ Es mag sein, dass dieses Patent aufgrund seines erst 14 Jahre späteren Erscheinens im September 1946 weniger bekannt ist, jedoch konnte kein früheres Patent aus der Radio- und Funktechnik mit einer expliziten Erwähnung, geschweige denn einer Definition des Begriffs gefunden werden. Hayes Gerät und damit der Begriff entstand während seiner Arbeit am Naval Research Laboratory (NRL) der US-Marine, wo er aktive und passive Sonargeräte für

⁸⁸ Als Beispiel sei hier auf Andreas Thurows Interton-Firmengeschichte verwiesen, vgl. Thurow, *Geschichte*.

⁸⁹ Eintrag „*Transceiver*“. In: *Wikipedia. Die freie Enzyklopädie*. Hrsg. von Wikimedia Foundation Inc. 29. Jan. 2015. URL: <https://de.wikipedia.org/w/index.php?title=Transceiver&oldid=138302071> (besucht am 24. 09. 2015).

⁹⁰ Im verlinkten Eintrag liest man im Abschnitt „Origin“ lediglich: „1930s: blend of transmitter and receiver.“ Eintrag „*transceiver*“. In: *Oxford English Dictionary*. Hrsg. von Oxford University Press. URL: <https://www.oxforddictionaries.com/definition/english/transceiver> (besucht am 24. 09. 2015).

⁹¹ Wikimedia Foundation Inc., Eintrag „*Transceiver*“.

⁹² So der Titel eines Patents von Alexander Nyman. „Combined Wireless Sending and Receiving System“. US-Pat. 1 606 775. 16. Nov. 1926. Vergleichbare Radio- und Funkpatente aus den 1920er und 1930er Jahren formulieren ähnlich.

⁹³ Harvey C. Hayes. „Directive Transceiver for Sound“. US-Pat. 2 406 767. 3. Sep. 1946, Spalte 9.

den Einsatz auf Schiffen entwickelte.⁹⁴ Sein Patent ist nicht nur medienhistorisch, sondern geradeheraus von medienarchäologischer Relevanz, verweist es doch treffend auf die *sonische* Verbundenheit akustischer und elektromagnetischen Wellen.⁹⁵ Radio bzw. (Rund-)Funk und Sonar müssen am Beispiel des Transceivers epistemologisch zusammengedacht werden. Darauf verweist Wolfgang Hagen medienhistorisch, wenn er die Bedeutung des Untergangs der Titanic im Jahr 1912 für den US-amerikanischen „Radio-Act“ aus dem gleichen Jahr unterstreicht. Darin wurden „die Frequenzbereiche, auf denen die Amateure senden, von denen [abgetrennt], auf denen die Schifffahrt generell und davon geschieden das Militär, sprich die Navy, ausschließlich senden durfte.“⁹⁶ Der „Radio-Act“ regelte Sende- und Empfangskanäle juristisch, heutige integrierte Schaltungen wie Bus Transceiver tun dies für Adress- und Datenleitungen in Digitalcomputern auf hochtechnische Weise.

5.4 Fazit

Abschließend ist dieses Projekt als deutlicher Appell für mehr Praxis in einer Disziplin wie der Medienwissenschaft zu verstehen. Ihre Legitimität findet diese Forderung nicht nur durch die Praxis des Retrocomputings, sondern auch bei Martin Heidegger: „Das Zuhandene braucht seinen Zeugcharakter nicht zu verlieren, um ‚Objekt‘ einer Wissenschaft werden zu können. Die Modifikation des Seinsverständnisses scheint nicht notwendig konstitutiv zu sein für die Genesis des theoretischen Verhaltens ‚zu den Dingen‘.“⁹⁷

Anstelle ihre Argumente also ausschließlich aus Erzählungen und technischen Dokumentationen zu ziehen, tut eine ihren Namen Ernst nehmende Medienarchäologie gut daran, die Untersuchungen von technischen Medien ganz wie im Retrocomputing des Öfteren auch direkt am, mit und vor allem *im* technischen Objekt durchzuführen. Kittler sprach sich dezidiert für eine Erweiterung des Textkorpus' aus. Weder Medienarchäologie noch eine vermeintlich traditionellere Mediengeschichte gewinnt auf lange Sicht etwas vom erneuten Verschließen vor den Quellen und Methoden der jeweils anderen Teildisziplin.

94 Vgl. Fred T. Erskine. „Harvey C. Hayes: First Superintendent of the Sound Division at the Naval Research Laboratory, Washington D. C.“ In: *Journal of the Acoustical Society of America* 137.4 (Apr. 2015), S. 2273.

95 Vgl. hierzu Wolfgang Ernst. „Zum Begriff des Sonischen (mit medienarchäologischem Ohr erhört/vernommen)“. In: *Das Sonische – Sounds zwischen Akustik und Ästhetik*. Hrsg. von Forschungszentrum Populäre Musik der HU Berlin. PopScriptum 10. 2008. URL: http://www2.hu-berlin.de/fpm/popscrip/themen/pst10/pst10_ernst.htm (besucht am 24. 09. 2015).

96 Wolfgang Hagen. *Das Radio. Zur Geschichte und Theorie des Hörfunks – Deutschland/USA*. München: Fink, 2005, S. 180 ff., Zitat auf S. 182.

97 Martin Heidegger. *Sein und Zeit*. 19. Aufl. [Erstausgabe: Halle, 1927]. Tübingen: Niemeyer, 2006, S. 361. Für eine dezidiert medienwissenschaftliche Reformulierung desselben Arguments, siehe den Abschnitt „Mediales Zeug als ‚um ...zu‘“ in Wolfgang Ernst. *Gleichursprünglichkeit. Zeitwesen und Zeitgegebenheit von Medien*. Berlin: Kadmos, 2012, S. 317 f.

Abbildungsverzeichnis

Übernommene Abbildungen anderer Autoren sind mit einer entsprechenden Quellenangabe versehen. Alle übrigen Abbildungen wurden eigenhändig erstellt.

1	Anzeigefehler im entwickelten Spiel auf der Originalhardware. Detailansicht aus einem Videostill.	3
2	Zwei exemplarische Original-Spielmodule der VC 4000.	8
3	Die Verschaltung der CPU (<i>S[ignetics]2650</i>) und des PVI (<i>S2636</i>) mit dem Modulschacht (Ausschnitt). Aus: Manfred Schneider. <i>Schaltpläne der VC4000</i> . Archivdatei mit PDF-Zeichnungen. Feb. 2007. URL: http://vc4000.de/vc4000_Schaltplaene.rar (besucht am 24.09.2015), Datei <i>vc4000-Mainboard.pdf</i>	9
4	Signalleitungen am Modulschacht. Der Signaltyp in der dritten Spalte ist aus der Perspektive des Spielmoduls angegeben.	10
5	Testaufbau der seriellen Schreibschnittstelle zwischen SRAM und Arduino auf einem Steckbrett.	14
6	Schaltung auf einem Standard-Spielmodul für die VC 4000 mit ROM-IC. Aus: Manfred Schneider. <i>Schaltpläne der VC4000</i> . Archivdatei mit PDF-Zeichnungen. Feb. 2007. URL: http://vc4000.de/vc4000_Schaltplaene.rar (besucht am 24.09.2015), Datei <i>2k-4krom-cartridge.pdf</i>	16
7	Beginn des Spielcodes von „Jagd“. Links: Spielcode aus der Binärdatei, rechts: Detailansicht einer fotografischen Aufnahme vom Display des Logikanalysators, nachträglich farbinvertiert und in s/w umgewandelt.	20
8	Verschaltung des <i>Output-Enable</i> -Eingangs am Datenbus-Transceiver; Wahrheitstabelle für ein OR-Gatter.	22
9	Der erste funktionierende Prototyp. Links: Ausschnitt aus einem Videostill, rechts: Platine mit manuellen Änderungen.	24
10	Erzeugung eines Schreibsignals für den \overline{WE} -Eingang des SRAMs; Wahrheitstabelle für ein NAND-Gatter	25
11	Schreibzyklus des <i>Signetics 2650</i> . Aus: Signetics Corporation, Hrsg. <i>Signetics 2650 Microprocessor</i> . Datenblatt. USA, 1975, S. 36, Abb. 11.	26
12	Schaltplan des „Schach-2“-Spielmoduls. Aus: Manfred Schneider. <i>Schaltpläne der VC4000</i> . Archivdatei mit PDF-Zeichnungen. Feb. 2007. URL: http://vc4000.de/vc4000_Schaltplaene.rar (besucht am 24.09.2015), Datei <i>6krom-1kram-cartridge.pdf</i>	28
13	Spiegelung des SRAM-Speichers auf dem Spielmodul „Schach 2“.	29
14	Ansicht der aktuellen Version (dritter Prototyp).	31
15	Schaltplan des entwickelten Multi-ROM-Moduls.	32
16	Grafischer Export des aktuellen Platinenlayouts aus KiCad.	33

Literatur

- Arduino, Hrsg. *Digital Pins*. Online-Dokumentation. URL: <https://www.arduino.cc/en/Tutorial/DigitalPins> (besucht am 24.09.2015).
- Hrsg. *Getting Started with Arduino*. Online-Dokumentation. URL: <https://www.arduino.cc/en/Guide/HomePage> (besucht am 24.09.2015).
- Camenzind, Hans. *Designing Analog Chips*. 2. Aufl. College Station/TX: Virtualbookworm Publishing, 2005. URL: http://www.designinganalogchips.com/_count/designinganalogchips.pdf (besucht am 24.09.2015).
- Ernst, Wolfgang. „Zum Begriff des Sonischen (mit medienarchäologischem Ohr erhört/vernommen)“. In: *Das Sonische – Sounds zwischen Akustik und Ästhetik*. Hrsg. von Forschungszentrum Populäre Musik der HU Berlin. PopScriptum 10. 2008. URL: http://www2.huberlin.de/fpm/popscrip/themen/pst10/pst10_ernst.htm (besucht am 24.09.2015).
- “Media Archaeography. Method and Machine versus History and Narrative of Media”. In: *Media Archaeology. Approaches, Applications and Implications*. Hrsg. von Erkki Huhtamo und Jussi Parikka. Berkeley, Los Angeles und London: University of California Press, 2011, S. 239–255.
- *Gleichursprünglichkeit. Zeitwesen und Zeitgegebenheit von Medien*. Berlin: Kadmos, 2012.
- Erskine, Fred T. “Harvey C. Hayes: First Superintendent of the Sound Division at the Naval Research Laboratory, Washington D. C.” In: *Journal of the Acoustical Society of America* 137.4 (Apr. 2015), S. 2273.
- Fairchild Semiconductor Corporation, Hrsg. *An Introduction to and Comparison of 74HCT TTL Compatible CMOS Logic*. Application Note 368. März 1984. URL: <https://www.fairchildsemi.com/application-notes/AN/AN-368.pdf> (besucht am 24.09.2015).
- Felzmann, Sebastian. „Been there, done that. Mediennostalgie als kreative Praxis zur Schaffung neuer retroider Spiele“. In: *SHIFT – RESTORE – ESCAPE. Retrocomputing und Computerarchäologie*. Hrsg. von Stefan Höltgen. Winnenden: CSW, 2014, S. 25–39.
- Forster, W. *Interton VC-4000 Video Computer*. Eintrag zur VC 4000 auf der Pong Picture Page. 27. Nov. 2005. URL: http://www.pong-picture-page.de/catalog/product_info.php?manufacturers_id=78&products_id=110 (besucht am 24.09.2015).
- Hagen, Wolfgang. *Das Radio. Zur Geschichte und Theorie des Hörfunks – Deutschland/USA*. München: Fink, 2005.
- Hayes, Harvey C. “Directive Transceiver for Sound”. US-Pat. 2 406 767. 3. Sep. 1946.
- Heidegger, Martin. *Sein und Zeit*. 19. Aufl. [Erstausgabe: Halle, 1927]. Tübingen: Niemeyer, 2006.
- Höltgen, Stefan. „OPEN_. Ein Vorwort zur Computerarchäologie“. In: *SHIFT – RESTORE – ESCAPE. Retrocomputing und Computerarchäologie*. Hrsg. von dems. Winnenden: CSW, 2014, S. 11–23.
- Hrsg. *SHIFT – RESTORE – ESCAPE. Retrocomputing und Computerarchäologie*. Winnenden: CSW, 2014.
- „Die NOPs und HALTs digitaler Medien. Programmierlehre maschinennaher Sprachen für Medienwissenschaftler“. In: *Grundlagenstudien aus Kybernetik und Geisteswissenschaften*.

- Internationale Zeitschrift für Modellierung und Mathematisierung in den Humanwissenschaften* 55.4 (Dez. 2014), S. 139–153.
- Horowitz, Paul und Winfield Hill. *The Art of Electronics*. 3. Aufl. New York: Cambridge University Press, 2015.
- Huhtamo, Erkki und Jussi Parikka, Hrsg. *Media Archaeology. Approaches, Applications and Implications*. Berkeley, Los Angeles und London: University of California Press, 2011.
- Jacobs, James. *WinArcadia*. Programm-Quellcode. Version 24.21. URL: <http://amigan.1emu.net/releases/WinArcadia-src.rar> (besucht am 24.09.2015).
- *Interton VC 4000 Coding Guide*. Allgemeine Hinweise zur Codestruktur von VC-4000-Spielen. Erstellt am 7. Feb 2007. 10. Feb. 2014. URL: <http://amigan.yatho.com/i-coding.txt> (besucht am 24.09.2015).
- *The sound issue will be fixed*. Kommentar auf Facebook in der geschlossenen Gruppe „Signetics 2650“. 24. Sep. 2014. URL: https://www.facebook.com/groups/392123604238327/permalink/643611902422828/?comment_id=643746732409345&offset=0&total_comments=8&comment_tracking=%7B%22tn%22:%22R1%22%7D (besucht am 24.09.2015).
- Keller, Mario und Johannes Maibaum. *VC4000 MultiRom Modul*. Code-Repository. URL: <https://github.com/mkeller0815/flappy4000/tree/master/MultiRom> (besucht am 24.09.2015).
- Kernighan, Brian W. und Dennis M. Ritchie. *The C Programming Language*. 2. Aufl. Englewood Cliffs/NJ: Prentice Hall, 1988.
- KiCad. *A Cross Plattform and Open Source Electronics Design Automation Suite*. Website des verwendeten Platinenlayoutprogramms. URL: <http://kicad-pcb.org> (besucht am 24.09.2015).
- Kittler, Friedrich. *Draculas Vermächtnis. Technische Schriften*. Leipzig: Reclam, 1993.
- „Protected Mode“. In: Ders. *Draculas Vermächtnis. Technische Schriften*. Leipzig: Reclam, 1993, S. 208–224.
- Klaß, Christian. „Twittern mit dem C64“. In: *Golem.de* (15. Juni 2009). URL: <http://www.golem.de/0906/67762.html> (besucht am 24.09.2015).
- Kuhn, Markus. *Standardized units for use in information technology or “What is a Megabyte ...?”* 19. Juli 1999. URL: <https://www.cl.cam.ac.uk/~mgk25/information-units.txt> (besucht am 24.09.2015).
- Kushner, David. “The Making of Arduino. How five friends engineered a small circuit board that’s taking the DIY world by storm”. In: *IEEE Spectrum* (26. Okt. 2011). URL: <http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino> (besucht am 24.09.2015).
- Lancaster, Don. *TTL Cookbook*. 25. Aufl. [Erstaufgabe 1974]. Carmel/IN: SAMS, 1994.
- Loebel, Jens-Martin. *Lost in Translation. Leistungsfähigkeit, Einsatz und Grenzen von Emulatoren bei der Langzeitbewahrung digitaler multimedialer Objekte am Beispiel von Computerspielen*. Glückstadt: Hülsbusch, 2014.
- Mahoney, Michael S. “The histories of computing(s)”. In: *Interdisciplinary Science Reviews* 30.2 (2005), S. 119–135.
- “2650 Now Being Shipped And Sampled”. In: *Microcomputer Digest* 2.1 (Juli 1975). Hrsg. von Microcomputer Associates Inc., S. 1–3.

- Morozov, Evgeny. "Open and Closed". In: *New York Times* (16. März 2013). URL: <http://www.nytimes.com/2013/03/17/opinion/sunday/morozov-open-and-closed.html> (besucht am 24.09.2015).
- "The Meme Hustler. Tim O'Reilly's crazy talk". In: *The Baffler* 22 (2013), S. 66–75. URL: <http://thebaffler.com/articles/the-meme-hustler> (besucht am 24.09.2015).
- „Die Rede von der Innovation. Der Jargon des Silicon Valley fängt sie alle ein“. Aus dem Englischen übers. von Matthias Fienbork. In: *Frankfurter Allgemeine Zeitung* (18. Apr. 2014). URL: <http://www.faz.net/-hbj-7ofqj> (besucht am 24.09.2015).
- MOS Technology Inc., Hrsg. *F2114/2114L. 1024 x 4 Static RAM*. Datenblatt. Ohne Datierung. URL: <https://www.princeton.edu/~mae412/HANDOUTS/Datasheets/2114.pdf> (besucht am 24.09.2015).
- Neumann, John von. "First Draft of a Report on the EDVAC". In: *IEEE Annals of the History of Computing* 15.3 (1993), S. 27–75.
- NXP Semiconductors, Hrsg. *74HC00; 74HCT00. Quad 2-input NAND gate*. 6. Rev. Datenblatt. 14. Dez. 2011. URL: http://www.nxp.com/documents/data_sheet/74HC_HCT00.pdf (besucht am 24.09.2015).
- Hrsg. *74HC32; 74HCT32. Quad 2-input OR gate*. 5. Rev. Datenblatt. 4. Sep. 2012. URL: http://www.nxp.com/documents/data_sheet/74HC_HCT32.pdf (besucht am 24.09.2015).
- Hrsg. *74HC595; 74HCT595. 8-bit serial-in, serial or parallel-out shift register with output latches; 3-state*. 7. Rev. Datenblatt. 26. Jan. 2015. URL: http://www.nxp.com/documents/data_sheet/74HC_HCT595.pdf (besucht am 24.09.2015).
- Nyman, Alexander. "Combined Wireless Sending and Receiving System". US-Pat. 1 606 775. 16. Nov. 1926.
- Eintrag "transceiver". In: *Oxford English Dictionary*. Hrsg. von Oxford University Press. URL: <https://www.oxforddictionaries.com/definition/english/transceiver> (besucht am 24.09.2015).
- Philips Semiconductors, Hrsg. *74HC245; 74HCT245. Octal bus transceiver; 3-state*. 3. Rev. Datenblatt. 31. Jan. 2005. URL: http://www.nxp.com/documents/data_sheet/74HC_HCT245.pdf (besucht am 24.09.2015).
- Pias, Claus. „Friedrich Kittler und der ‚Mißbrauch von Heeresgerät‘. Zur Situation eines Denkbildes 1964 – 1984 – 2014“. In: *Merkur. Deutsche Zeitschrift für europäisches Denken* 69.791 (Apr. 2015), S. 31–44.
- Ross, Kevin. "Basic Circuits. Bypass Capacitors". In: *Encoder. The Newsletter of the Seattle Robotics Society* (Juni 1997). URL: <http://www.seattlerobotics.org/encoder/jun97/basics.html> (besucht am 24.09.2015).
- Schneider, Manfred. *Schaltpläne der VC4000*. Archivdatei mit PDF-Zeichnungen. Feb. 2007. URL: http://vc4000.de/vc4000_Schaltplaene.rar (besucht am 24.09.2015).
- Signetics Corporation, Hrsg. *Signetics 2650 Microprocessor*. Datenblatt. USA, 1975.
- Silicon Valley Historical Association. *NXP Semiconductors*. 2008. URL: <http://www.siliconvalleyhistorical.org/nxp-semiconductors-history> (besucht am 24.09.2015).
- Singular, Soci und The Dreams (Pseudonyme). *Singular browser*. URL: <http://singularcrew.hu/browser> (besucht am 24.09.2015).

- Stallman, Richard. *The GNU Manifesto*. Mehrfach überarbeitete Version auf gnu.org. (Erstveröffentlichung 1985). 12. Juni 2015. URL: <https://www.gnu.org/gnu/manifesto> (besucht am 24. 09. 2015).
- Takhteyev, Yuri und Quinn DuPont. "Retrocomputing as preservation and remix". In: *Library Hi Tech* 31.2 (2013), S. 355–370.
- Texas Instruments Inc., Hrsg. *SN74LVC125A. Quadruple Bus Buffer Gate With 3-State Outputs*. Datenblatt. Dallas/TX, Feb. 2004. URL: <http://pdf1.alldatasheet.com/datasheet-pdf/view/171748/TI/LVC125A.html> (besucht am 24. 09. 2015).
- Thurrow, Andreas. *Geschichte*. In: VC4000.de. Ohne Datierung. URL: <http://vc4000.de/html/geschichte.html> (besucht am 24. 09. 2015).
- *Spiele*. In: VC4000.de. Ohne Datierung. URL: http://vc4000.de/html/body_spiele.html (besucht am 24. 09. 2015).
- Utron Technology Inc., Hrsg. *UT6264C. 8K X 8 Bit Low Power CMOS SRAM*. 1.4. Datenblatt. Hsinchu, 15. Mai 2003. URL: <http://cdn-reichel.de/documents/datenblatt/A300/UT6264.PDF> (besucht am 24. 09. 2015).
- Valvo, Hrsg. *Signetics Programmable Video Interface (PVI) 2636*. Datenblatt. Ohne Datierung.
- Vogt, Patrick. *Interton VC4000 Homebrew-EEPROM-Modul?* Beitrag im Online-Forum *circuit-board.de*. 10. Apr. 2014. URL: <http://circuit-board.de/forum/index.php/Thread/12776-Interton-VC4000-Homebrew-EEPROM-Modul/?postID=324054#post324054> (besucht am 24. 09. 2015).
- Ward, Jack. "The 555 Timer IC. An Interview with Hans Camenzind—The Designer of the Most Successful Integrated Circuit Ever Developed". In: *Transistormuseum.com* (Juni 2004). URL: http://semiconductormuseum.com/Transistors/LectureHall/Camenzind/Camenzind_Index.htm (besucht am 24. 09. 2015).
- Wikimedia Foundation Inc., Hrsg. *Wikipedia. Die freie Enzyklopädie*.
- Eintrag „Interton VC4000“*. In: *Wikipedia. Die freie Enzyklopädie*. Hrsg. von Wikimedia Foundation Inc. 15. März 2015. URL: https://de.wikipedia.org/w/index.php?title=Interton_VC4000&oldid=139806399 (besucht am 24. 09. 2015).
- Eintrag „Transceiver“*. In: *Wikipedia. Die freie Enzyklopädie*. Hrsg. von Wikimedia Foundation Inc. 29. Jan. 2015. URL: <https://de.wikipedia.org/w/index.php?title=Transceiver&oldid=138302071> (besucht am 24. 09. 2015).