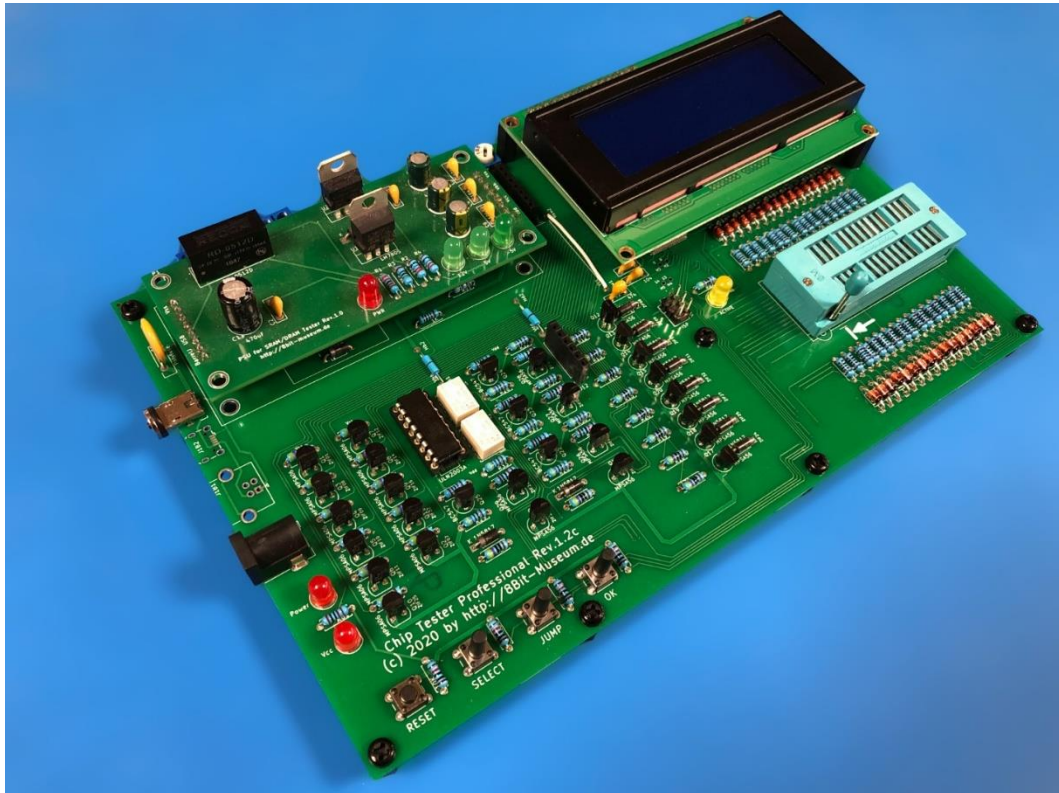


Retro Chip Tester Professional

BENUTZERHANDBUCH



Rev.1.2

Hinweis:

Dieses Handbuch enthält bereits Erklärungen zu Features, die erst mit dem nächsten Firmware-Release verfügbar sein werden.

Diese Seite wurde absichtlich leer gelassen

Inhaltsverzeichnis

1	Allgemeines zum Retro Chip Tester Professional	9
1.1	Übersicht.....	10
2	Bedienung	17
2.1	Tastenbelegung	17
2.2	Das Menü.....	19
2.3	Suchen eines ICs / Unterstützte ICs	27
3	Testen von Chips	29
3.1	Testen von SRAMs und DRAMs	29
3.1.1	Einsetzen von Chips	29
3.1.2	Auswahl des korrekten Bausteins	30
3.1.3	Ausgabe des Testergebnis	31
3.1.4	Speichern des Testergebnisses	34
3.1.5	Identifizieren von Page Mode, Nibble Mode und Static Column Mode bei DRAMs	38
3.2	Spezielle Speicherbausteine	39
3.2.1	Testen von SIMM und SIPP Modulen	39
3.2.2	Testen von ZIP Speicher	43
3.2.3	Testen von 7481 bzw. 7484 bipolar SRAM	46
3.2.4	Testen von MK4008 und MK4006	47
3.2.5	Testen von Intel 2107 und kompatible DRAMs	49
3.2.6	Testen von Intel D8155 und D8156	50
3.2.7	Testen von Motorola MCM6605 und kompatible DRAMs	50
3.2.8	Testen von Intel 1101	51
3.2.9	Testen von TMS 4050 und TMS 4051	52
3.2.10	Selbstdefinierte SRAM, DRAM und ROM Speicherbausteine	53
3.3	Problematische Speicherbausteine	58
3.3.1	Testen von älteren DRAMs	58
3.3.2	Testen von ICs mit einer erhöhten Stromaufnahme	59
3.3.3	Testen der Ausgangstreiber von ICs	59
3.3.4	Verwenden des Entkopplungsadapters (Decoupling Adapter)	60
3.3.5	Verwenden des „on-board“ Entkopplungskondensators (ab Rev.1.2k)	62
3.3.6	Testen von ITT4027 (und ggf. kompatible SRAMs)	63
3.3.7	Testen von SIMM/SIPP Modulen	64
3.3.8	DRAMs aus russischer bzw. osteuropäischer Fertigung	64
3.4	Testen von ROM, PROM, EPROM oder EEPROM	65
3.4.1	Identifizieren von Inhalten	66
3.4.2	Auslesen von ROM, PROM, EPROM oder EEPROMs	67
3.4.3	Vergrößern der internen ROM-Datenbank	68
3.4.4	Verwenden von vorbereiteten Datenbanken	68
3.4.5	Besonderheiten beim Auslesen von ROMs und PROMs	70
3.4.6	Auslesen von 2716 (2k x 8), 2532 (4k x 8), 2508 (1k x 8), 2758 (1k x 8)	74
3.4.7	Auslesen eines Intel 1702A / 1602A	75
3.4.8	Auslesen des Character ROM RO-3-2513 bzw. der Motorola MCM6670-Serie	77
3.4.9	Auslesen des Character ROM 2513 (+5V/-5V/-12V Version)	78
3.4.10	Auslesen der ROMs MM4204 / MM5204	78

3.4.11	Auslesen des ROM Signetics 2530	79
3.4.12	Auslesen des ROM Mostek MK28000 und TMS4800	81
3.4.13	Auslesen des EEPROM uPD454 und ROM uPD464	82
3.4.14	Auslesen des EPROM CDP18U42	82
3.4.15	Auslesen von Commodore C64 Cartridges	83
3.4.16	Auslesen von Commodore VC20 Cartridges	84
3.4.17	Auslesen von Atari VCS/2600 Cartridges	85
3.4.18	Auslesen von MSX Cartridges	86
3.4.19	Auslesen von 1-16 Mbit EPROMs	87
3.5	Testen von PLA und FPLA	89
3.5.1	Besonderheiten bei Commodore PLAs	90
3.6	Lesen der „elektronischen Signatur“ eines EPROMs	92
3.7	Testen von Logik-ICs (74xx, 40xx und weitere)	94
3.7.1	Chips mit besonderer Spannungsversorgung	95
3.7.2	Pinbelegung von Logik-Chips	97
3.7.3	Testen von russischen Chips	101
3.7.4	Testen von alten bipolaren Logik-Chips und S-TTL/H-TTL	102
3.7.5	Einsatz des Experimentier-/Breakout-Adapters	107
3.7.6	Testen von Multivibratoren (74121, 74122, 74123 et. al.)	109
3.7.7	Testen von Multivibratoren (CD4098, CD4528, CD4538, DM9602)	111
3.7.8	Testen von erweiterbaren Logik-ICs („Expander“)	112
3.7.9	Identifizieren unbekannter Logik-ICs	113
3.8	Testen von sonstigen Bausteinen	114
3.8.1	Visual Fuzzy Test	115
3.8.2	Testen von Joysticks	116
3.9	Auslesen von PALs (Programmable Array Logic)	117
3.9.1	Vorbereitung	118
3.9.2	Starten des Auslesevorgangs	120
3.9.3	Weitere Möglichkeiten für den Einsatz mit Beispiel	121
3.9.4	Alternative Möglichkeit PALs und GALs zu rekonstruieren	123
3.10	Testen von SOIC (SOP) Bausteinen	124
3.11	Einfacher Taktgenerator	125
3.11.1	Takterzeugung: nicht-überlappender Takt bzw. Quadratur-Takt	126
3.12	Einfacher Frequenzzähler	127
4	Programmieren von EPROMs	129
4.1	Einstellen der Programmierspannung (Vpp)	131
4.2	Auswahl der zu programmierenden ROM-Datei (ab FW v.23)	132
4.3	2708 EPROM	133
4.4	2716/2516 EPROM and Hitachi HN48016 EEPROM	135
4.5	TMS2716 EPROM (ab FW v.22)	137
4.6	2532 EPROM	138
4.7	2732 EPROM	139
4.8	2564 EPROM (ab FW v.22)	140
5	Einstellungsmenü / Konfigurationsmenü	141
6	Warnhinweise	144
7	Bekannte Probleme und Einschränkungen	145
7.1	Erkennung von Fehlern (Speicherchips)	145
7.2	Testen von “non-volatile static RAM” (NVRAM) (“Zeropower” RAMs)	152
7.3	Erkennen von Fehlern (Logik- und Speicherbausteine)	153

7.3.1	Ungültige Signalpegel.....	153
7.3.2	Defekte Eingangs- bzw. Ausgangstreiber	154
7.3.3	Timing-Probleme	154
7.3.4	„Falsche“ Ergebnisse.....	155
7.3.5	Weitere Probleme	155
7.4	"In circuit" Tests	156
7.5	Verwenden einer SD-Karte.....	157
8	Anhang: PAL Analyse und Synthese mit dem RCT	158
9	Anhang: Ein paar Tipps.....	169
9.1	Verwenden eines 4164 oder 41256 als 4116.....	169
9.2	Verwenden eines 2716 als 2708	170

Diese Seite wurde absichtlich leer gelassen

Sicherheitshinweise:



Bitte beachten sie folgende Hinweise:

- Lassen Sie das Gerät im Betriebszustand niemals unbeaufsichtigt.
- Wird das Gerät nicht verwendet, lassen sie es nicht an der Spannungsversorgung angeschlossen.
- Wenn eine ungewöhnliche Geruchsentwicklung oder Hitzeentwicklung am Gerät festzustellen ist, schalten sie das Gerät sofort aus.
- Die Linearregler (7805, 7905) können je nach Belastung bei längerem Betrieb heiß werden. Eine Überlastung kann zu einer übermäßigen Erhitzung der Linearregler führen.
- Stellen Sie sicher, dass das Gerät an einem Ort außerhalb der Reichweite von direkten Wärmequellen, offenen Flammen und brennbaren Materialien und Flüssigkeiten aufgestellt wird. Das Gerät ist für den Betrieb bei Raumtemperatur vorgesehen.
- Das Gerät ist für den USB-Betrieb und einer stabilisierten Versorgungsspannung von maximal 9V vorgesehen. Verwenden Sie nur geprüfte und geeignete Netzteile zur Stromversorgung. Achten Sie darauf, dass die Spannungsquelle den Anforderungen entspricht.
- Komponenten dürfen ausschließlich im Rahmen der angegebenen Spannungs- und Stromwerte getestet werden. Die Grenzen des Produkts sind im entsprechenden Datenblatt aufgeführt.
- Achten Sie darauf, dass keine leitenden Objekte die Platine berühren bzw. Kurzschlüsse verursachen.
- Das Produkt ist empfindlich gegenüber elektrostatischer Entladung (ESD), die elektronische Komponenten beschädigen oder zerstören kann.

Der Tester ist für technisch versierte Nutzer wie Elektroniker, Ingenieure, Entwickler und Studenten bestimmt, die Speicher und Logikkomponenten testen möchten. Es ist nicht für die Verwendung als fertiges Verbraucherprodukt vorgesehen. Das Produkt ist nicht für industrielle Anwendungen oder sicherheitsrelevante Zwecke geeignet. Die Verwendung in Verbindung mit Medizingeräten oder in der Luft- und Raumfahrt ist ebenfalls nicht zulässig.

Der Speichertester wurde bereits mehrfach aufgebaut und funktioniert beim Entwickler. Das heißt aber nicht unbedingt, dass das automatisch auch bei einem Nachbau der Fall ist. Das Dokument soll helfen den Aufbau und die Funktionsweise zu erklären und gibt Hinweise, wie der Speichertester nachgebaut werden kann. Es werden oft zusätzliche, technische Informationen gegeben. Nicht alles ist für das Verständnis unbedingt notwendig. Bei Problemen können diese aber hilfreich sein.

Wer den Tester nachbaut, der ist selbst für die richtige Auswahl der Bauteile und den Aufbau verantwortlich. Eine Garantie für eine korrekte Funktion wird nicht gegeben. Es wird keine Haftung dafür übernommen, wenn beteiligte Geräte durch den Einsatz des Speichertesters Schaden nehmen. In dem Zusammenhang bitte unbedingt auch die Warnhinweise im Abschnitt 6 beachten.

Einige Abschnitte sind mit einem „DANGER“ gekennzeichnet. Das heißt jetzt nicht wirklich, dass der Inhalt „gefährlich“ ist, sondern, dass hier der Tester beschädigt werden kann.



Warnhinweise sind entsprechend gekennzeichnet.



Urheberrecht und Gewährleistung

© Soft- und Hardware: Stephan Slabihoud, 8Bit-Museum.de

Alle Rechte vorbehalten. Kein Teil dieses Handbuchs, einschließlich der darin beschriebenen Produkte und Software, darf ohne ausdrückliche schriftliche Genehmigung des Entwicklers in irgendeiner Form oder auf irgendeine Weise reproduziert, übertragen oder kommerziell verwendet werden.

Die RCT-Leiterplatte (bestückt mit dem ATmega2560 oder unbestückt ohne Bauteile („Blank Board“)), die Firmware und alle zusätzlichen Leiterplatten (z. B. Adapterboards) werden im Folgenden als „Produkte“ bezeichnet.

Die Verwendung der Hard- und Software geschieht auf eigene Gefahr. Es wird keine Haftung für etwaige Schäden übernommen, die beim Einsatz der vorliegenden Soft- und Hardware entstehen könnten. Es wird keine Gewährleistung für die korrekte Funktionsfähigkeit der bereitgestellten Funktionen übernommen. Es besteht auch kein Anspruch auf eine Fehlerbehebung, auch wenn der Entwickler bemüht ist Fehler schnellstmöglich zu beseitigen.

Wenn die Produkte außerhalb der EU gekauft werden, gilt diese Gewährleistung für ein Jahr ab Verkaufsdatum. Wenn die Produkte in der EU gekauft werden, gilt die Gewährleistung zwei Jahre ab Verkaufsdatum. Alle Komponenten, die vom Benutzer dazugekauft werden müssen, um ein funktionierendes RCT zu bauen, sind nicht von dieser Gewährleistung abgedeckt. Ebenso wird keine Gewährleistung für ein fertig montiertes RCT übernommen, mit Ausnahme der gelieferten Leiterplatten.

Der Entwickler haftet nicht für Mängel, die durch Nichteinhaltung von Vorgaben, Missbrauch oder falsche Bedienung durch den Benutzer verursacht wurden, einschließlich unsachgemäßer Installation oder Prüfungen oder wenn das Produkt in irgendeiner Weise geändert oder modifiziert wurde. Diese eingeschränkte Gewährleistung ist das einzige und ausschließliche Rechtsmittel des Endbenutzers gegenüber dem Entwickler, sofern gesetzlich zulässig. Die Produkte werden „wie besehen“ und „mit allen Fehlern“ bereitgestellt. Der Entwickler lehnt alle anderen ausdrücklichen oder stillschweigenden Gewährleistungen der Marktgängigkeit für einen bestimmten Zweck ab.

Die Produkte wurden entwickelt, um unterstützte ICs in alten Heimcomputern, Synthesizern, Arcade- und Flipperautomaten zu testen. Es ist nur die nicht-kommerzielle Nutzung im privaten Kontext vorgesehen. Die Produkte sind nicht für die Verwendung an sicherheitskritischen Stellen zugelassen, in denen bereits erwartet werden kann, dass ein Versagen des Produkts schwere Verletzungen oder den Tod in irgendeiner Form verursacht. Der Benutzer erkennt an und stimmt zu, dass jede andere Verwendung dieser Produkte ausschließlich auf Risiko des Benutzers erfolgt und dass der Benutzer allein für die Einhaltung aller gesetzlichen und behördlichen Anforderungen in Verbindung mit einer solchen Verwendung verantwortlich ist.

Verzicht auf Folgeschäden. In keinem Fall haftet der Entwickler gegenüber dem Benutzer oder Dritten für besondere, begleitende, indirekte, strafbare, zufällige Folgeschäden oder Schadensersatz im Zusammenhang mit oder aus den hier bereitgestellten Produkten, unabhängig davon, ob der Entwickler auf die Möglichkeit solcher Schäden hingewiesen hatte oder nicht. Dieser Abschnitt gilt auch nach Ablauf der Gewährleistungsfrist.

1 Allgemeines zum Retro Chip Tester Professional

Wer den Tester aufbaut, sollte sich zuerst einen Überblick über die inzwischen recht umfangreiche Dokumentation verschaffen.

DE-1 – Aufbau, Programmierung und Fehlersuche

Das Dokument beschreibt Aufbau, Programmierung und Fehlersuche.

Hierzu gibt es weitere Dateien:

- Ordner „Programming Step 1 - Set Fuses“
- Ordner „Programming Step 2 - Flash Firmware“

Es ist alles vorhanden, um den ATmega zu programmieren. Es ist i.d.R. nicht notwendig weitere Software auf dem Internet zu laden.

DE-2 – Anwenderhandbuch

Das Dokument beschreibt die Anwendung des RCT.

Es befinden sich weiterhin

- CRC32-Listen (zur Identifizierung von ROMs),
- Eigene IC-Definitionen (zur Definition sehr spezieller ICs),
- Vergleichslisten (damit die richtigen Testeinstellungen verwendet werden)

in den Unterordnern, in denen dieses Dokument gespeichert ist.

DE-3 – Stückliste (BOM)

Das Dokument (nicht öffentlich verfügbar) enthält eine Übersicht der benötigten Bauteile und Links auf vorbereitete Warenkörbe. Im selben Verzeichnis befinden sich auch Bauteillisten für Reichelt Elektronik und Digikey. Weiterhin sind dort interaktive Bestückungslayouts vorhanden.

1.1 Übersicht

Der Retro Chip Tester Pro wurde entwickelt, um „alte“ Speicherchips aus den 1970er und 1980er Jahren testen zu können, die von heutigen Programmiergeräten oft nicht mehr erkannt werden.

Zu den unterstützten Speicherchips gehören u.a. (zzgl. alle Pin-kompatible Typen):

SRAMs (Beispiele)	
Größe	Chips (Beispiele)
4x4	74170, 74670
4x8	TC4036, TC4039
8x2	74172
16x1	7481 / 7484 (mit Adapter)
16x4	3101, 7489, 74189, 74219
16x9	74F211, 74F311
16x12	74F213, 74F313
32x8	74LS218, 74LS318
64x1	4505
64x4	74C910
64x9	82S09, 93419
128x8	6810
256x1	8216, 2116, 8217, 2700, 2701, 3106, 3107, 93410, 1101
256x4	2101A, 2111, 2112A, 74920, 74921, 6561
256x8	81C50, 81C51, 81C52, GTE3539, 8155, 8156
256x9	82S10, 82S12
512x4	D2113
512x8	81C54
1k x 1	2102, 8102, 2115, 2511
1k x 4	2114, 2148, 2149, 4045, 5114, 6514, C214, U224
1k x 4	6550
1k x 8	4118, 4801, 8185
2k x 1	HM6303H/L
2k x 8	2016, 2416, 4016, 4802, 4812, 6116, 6512, TC5516
4k x 1	2147, MK4104
4k x 4	7C168, 6168, 5416, P4C168, P4C169
8k x 8	2064, 2464, 6264, 2465
8k x 9	P4C163
16k x 1	8167, 6267, 6167, 2167, P4C167
16k x 4	P4C188, P4C198
32k x 8	20256, 61256, 62256, 71256, 43256, 43257
32k x 9	CY7C188
64k x 1	P4C187
64k x 4	P4C1258, P4C1281, P4C1298
64k x 8	61512, 24512
128k x 8	621024, 431000, P4C1023, 6173081, 638100, 63C1024
128k x 9	CY7C1088
256k x 1	P4C1257
256k x 4	P4C1026
512k x 8	AS6C4008, P4C1048, F7447APC
1024k x 1	P4C107

FIFO RAMs (Beispiele)	
Größe	Chips (Beispiele)
16 x 4	40105, 74LS222, 74LS224, 74LS227, 74LS228, 74LS232
16 x 5	74S225, 74LS229, 74LS233
64 x 4	74LS234, 74LS236, 74F413, 74HC7403
64 x 5	74LS235
256 x 9	7200, 7400 PLCC
512 x 9	7201, 7401 PLCC
1k x 9	7202, 7402 PLCC
2k x 9	7203, 7403 PLCC
4k x 9	7204, 7404 PLCC

NOVRAMs (Beispiele)	
Größe	Chips (Beispiele)
64 x 4	X22C10
256 x 4	X22C12
512 x 8	X20C04, X20C05
2k x 8	X20C16, X20C17

DRAMs (Beispiele)	
Größe	Chips (Beispiele)
1k x 1	MK4008
4k x 1	2104A, MK4015, MK4027, 7027, 2107
8k x 1	4108-x0/x1, MK4115-x0/x1, 2108H/L, 5298A/B
8k x 4	4408NLT/NLB
16k x 1	4116, 2117, 6116, 8116, 416, 2116, 3716, U256
16k x 1	2118, K565RU6
16k x 4	4416, 2620
32k x 1	3732H (4532-L4), 3732L (4532-L3)
2x 16k x 1	4132 (Modul 1, Modul 2), MK4332 (Modul 1, Modul 2)
64k x 1	4164, 2600, K565RU5, 8264, 3764
64k x 4	4464, 41464, 50464
2x 64k x 1	41128 (Top Module, Bottom Module)
256k x 1	41256, 53256, 81256, MT1259
256k x 4	44256, 514256
1024k x 1	41024, 411000
1024k x 4	44400, 514400
256k x 8	256 kB SIMM / SIPP (30-pol.)
256k x 1	256 kB SIMM / SIPP (30-pol.) nur Parity
1024k x 8	1024 kB SIMM / SIPP (30-pol.)
1024k x 1	1024 kB SIMM / SIPP (30-pol.) nur Parity
64k x 4	ZIP 20
256k x 1	ZIP 16
256k x 4	ZIP 20
1024k x 1	ZIP 20
1024k x 4	ZIP 20

ROMs / PROMs / EPROMs (Beispiele)	
Typ	Chips (Beispiele)
2k x 8	6540
(P)ROM	2308, 2316A, 2316B, 2332, 2364, 23128, 23256, 23512, 231000/231001, 232000, 234000, RO-3-2513, 6670, 2513 (with adapter)
EPROMs	2704, 2708, 2716, TMS2716, 2732, 2764, 27128, 27256, 27512, 271001, 272001, 274001, 1702 (with adapter), 4204/5204 (with adapter), TMS2716, TMS2564, CDP18U42
C64	2x 8k Cartridges
VC20	8k bis 64k Cartridges
VCS/2600	2k, 4k, 8k, (16k) Cartridges

bipolare ROMs / PROMs (Beispiele)	
Typ	Chips (Beispiele)
32 x 8	7488, 74188, 74288
256 x 4	74187, 74287, 74387
256 x 8	74S271, 74S371, 74S470, 74S471
512 x 4	74S270, 74S370, 74S570, 74S571
512 x 8	74S472, 74S473, 74S474, 74S475
1k x 4	74S476, 74S477, 74S572, 74S573
1k x 8	74S478, 74S479
2k x 4	82S184, 82S185
2k x 8	82S190, 82S191
4k x 4	82S195
4k x 8	82S321
8k x 8	82S641

Logic Chips	
Type	Chips
74xxx	7400, 01, 01(H), 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 51(LS), 52, 53, 53(H), 54, 54(H), 54(LS), 55(H), 55(LS), 56, 57, 58, 60, 61, 62, 64, 65, 68, 69, 70, 71(H), 71(L), 72, 73, 74, 75, 76, 76(LS), 77, 78(H), 78(L), 78(LS), 79, 80, 82, 83, 85, 85(C), 86, 86(L), 87, 90, 91, 92, 93, 93(C), 93(L), 94, 95, 95(C), 95(L), 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 107(LS), 108, 109, 110, 111, 112, 113, 114, 115, 116, 118, 119, 120, 121, 122, 123, 125, 126, 128, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 210, 221, 230, 231, 237, 238, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 251, 253, 155, 256, 257, 258, 259, 260, 261, 265, 266, 269, 273, 275, 276, 278, 279, 280, 283, 284, 285, 286, 290, 292, 293, 294, 295, 298, 299, 310, 322, 323, 347, 340, 341, 344, 348, 350, 351, 352, 353, 354, 355, 356, 357, 363, 364, 365, 366, 367, 368, 373, 374, 375, 376, 377, 378, 379, 381, 381(F), 382, 382(F), 384, 385, 386, 388, 390, 393, 395, 396, 398, 399, 405(S), 412, 416(S), 422, 423, 425, 426, 432, 436, 437, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 456, 461, 465, 466, 467, 468, 490, 491, 518, 519, 520, 521, 522, 524, 526, 527, 528, 533, 534, 535, 536, 537, 538, 539, 540, 541, 543, 544, 545, 546, 547, 547(F), 548(F), 560, 561, 563, 564, 566, 567, 568, 569, 573(LS), 574, 575, 576, 577, 579, 580, 583, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 604, 605, 606, 607, 620, 621, 622, 623, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 651, 652, 653, 654, 657, 666, 667, 668, 669, 671, 672, 673, 674, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 696, 697, 698, 699, 720, 721, 740, 741, 744, 748, 756, 757, 758, 759, 760, 762, 763, 779, 793, 794, 795, 796, 797, 798, 800, 802, 804, 805, 808, 810, 821, 822, 823, 824, 825, 826, 827, 828, 832, 841, 842, 843, 844, 845, 846, 848, 850, 851, 857, 861, 862, 863, 864, 866, 867, 869, 873, 874, 876, 878, 879, 880, 881, 885, 900(ALS), 901(C), 902(ALS), 902(C), 903(ALS), 903(C), 904(C), 906(C), 914(C), 915(C), 940, 941, 990, 991, 992, 993, 994, 995, 996, 2540, 2541, 3037, 7001, 7002, 7014, 7032, 7266, 8541, 9034, 9035, 9114, 9115, 9134, 9135, 9240, 9244, 9245
75xxx	75121, 75122, 75123, 75124, 75125, 75127, 75138, 75140, 75154, 75160, 75172, 75173, 75189, 75361(, 75361, 75450, 75451, 75452, 75453, 75454, 75460, 75461, 75462, 75463, 75464, 75466, 75467, 75468, 75469, 75470, 75471, 75472, 75473, 75474, 75491, 75492, 75494, 75497, 75498

Logic Chips	
Type	Chips
4xxx	4000, 01, 02, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 38, 40, 41, 42, 43, 44, 48, 49, 50, 51, 52, 53, 54, 55, 56, 59, 60, 63, 66, 67, 68, 69, 70, 71, 72, 73, 75, 76, 77, 78, 81, 82, 85, 86, 93, 94, 95, 96, 97, 98, 99, 014, 032, 097, 098, 101, 102, 103, 104, 106, 107, 109, 110, 116, 147, 160, 161, 162, 163, 174, 175, 192, 193, 194, 240, 244, 245, 257, 373, 374, 511, 4316, 4428
45xx	4501, 02, 03, 04, 06, 07, 08, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 22, 26, 28, 29, 30, 31, 32, 38, 39, 43, 51, 55, 56, 58, 60, 61, 72, 74, 84, 85, 98, 4723, 4724, 4929, 4930, 4931, 4934, 4935, 49700, 49701, 49702, 49703, 49704, 49705, 49710, 49711, 49713, 49714, 49805
DDR DLxx	000, 002, 003, 004, 008, 010, 011, 014, 020, 021, 030, 032, 037, 038, 040, 051, 074, 086, 090, 093, 112, 123, 132, 155, 164, 175, 192, 193, 194, 251, 253, 257, 259, 374, 540, 541, 299
UdSSR 153-555	AP3, AP4, AP5, AP6, AP9, AP10(K1102), AP10, AP11P6(K1102), AP12P6(K1102), AP13P6(K1102), AP14P6(K1102), AP14, AP15, AP16, AP17, AP24, AP26, ID1, ID3, ID4, ID5, ID6, ID7, ID10, ID14, ID18, ID22, ID24, IE1, IE2, IE4, IE5, IE6, IE7, IE8, IE9, IE10, IE11, IE12, IE13, IE14, IE15, IE16, IE17, IE18, IE19, IE20, IK2, IM1, IM2, IM3, IM5, IM6, IM7, IP1(KR559), IP2(KR559), IP2, IP3, IP4, IP5, IP6, IP7, IP8, IP9, IR1, IR2, IR5, IR8, IR9, IR10, IR11, IR12, IR13, IR15, IR16, IR22, IR23, IR24, IR25, IR27, IR28, IR29, IR30, IR33, IR34, IR35, IR37, IR38, IR40, IR41, IR42, IR43, IR52, IV1, IV2, IV3, KP1, KP2, KP5, KP7, KP11, KP12, KP13, KP14, KP15, KP16, KP17, KP18, KP19, KP20, KT1(K1010), LA1, LA2, LA3, LA4, LA6, LA7, LA8, LA9, LA10, LA11, LA12, LA13, LA16, LA18P6(K1102), LA19, LD1, LE1, LE2, LE3, LE4, LE5, LE6, LE7, LE8, LE11, LI1, LI2, LI3, LI4, LI5P6(K1102), LI6, LI7, LI9, LL1, LL2P6(K1102), LL3, LN1, LN2, LN3, LN4, LN5, LN6, LN7, LP5, LP8, LP9, LP10, LP11, LP12, LP13, LP14, LR1, LR3, LR4, LR9, LR10, LR11, LR13, PP4, PR6, PR7, SP1, SP2, TL1, TL2, TL3, TM2, TM5, TM7, TM8, TM9, TM10, TR2, TV1, TV6, TV9, TV10, TV11, TV15
UdSSR 561 CMOS	ID1, ID6, ID7, ID23, IE4, IE5, IE8, IE9, IE10, IE11, IE14, IE16, IE19, IE20, IE21, IM1, IP2, IP6, IR2, IR6, IR9, IR14, KP1, KP2, KP3, KP4, KP5, KT1, KT3, LA7, LA8, LA9, LA10, LE5, LE6, LE10, LI2, LN1, LN2, LN3, LP1, LP2, LP4, LP14, LS2, PR1, PU2, PU3, PU4, PU6, PU7, PU9, SA1, TL1, TM2, TM3, TR2, TV1, UM1
Misc	DIS1417, DL1414, DL2416Rev1, HP730x, HP7340, TIL306/307, TIL308/309, TIL311, VQB76,

Logic Chips	
Type	Chips
Misc Logic 1	3404, 6529 MOS, 7707 MOS, 7708 MOS, 7709 MOS, 7711 MOS, 7712 MOS, 7713 MOS, 7714 MOS, 7715 MOS, 8708 MOS, 8713 MOS, 8090, 8091, 8092, 8093/7093, 8094/7094, 8095/80C95, 8096/80C96, 8097/80C97, 8098/80C98, 8121, 8123, 8131/7131, 8160/7160, 8200/7200, 8205/3205, 8212/3212, 8216/3216, 8219/82C19, 8220/7220, 8226/3226, 8230, 8231, 8233, 8234, 8235, 8241, 8242, 8250, 8251, 8252, 82S62, 8263, 8264, 8266, 8267, 8270, 8271, 8280, 8281, 8282, 8283, 8286, 8287, 8290, 8291, 8292, 8293, 8415A, 8455A, 8470A, 8471A, 8480A, 8481A, 8490A, 8530, 8532, 8533, 8555, 8556, 8560, 8563, 8570, 8590, 8640/7640, 8641/7641, 8808A, 8810/7810, 8811/7811, 8812/7812, 8815A, 8816A, 8819/7819, 8829A, 8830, 8833/7833, 8835/7835, 8836/7836, 8837/7837, 8838/7838, 8840A, 8848A, 8855A, 8870A, 8875A, 8880A, 8881A, 8885A, 8890A, 8891A, 9000, 9001, 9002, 9003, 9004, 9007, 9009, 9012, 9016, 9017, 9024, 9300, 9301, 9308, 9309, 9310, 9312, 9314, 9316, 9317, 9318, 9321, 9322, 9324, 9334, 93S47, 93S62, 9368, 9370, 8H16, 8H70, 8H80, 8H90, 8T09, 8T10, 8T13, 8T14, 8T22, 8T23, 8T24, 8T26, 8T28, 8T37, 8T38, 8T80, 8T90, 8T93, 8T94, 8T95, 8T96, 8T97, 8T98, 8T125, 8T245, Am25LS07, Am25LS08, Am25LS09, Am25S10, Am25LS14, Am25LS15, Am25S18, Am25LS22, Am25LS23, Am25LS2518, Am25LS2519, Am25LS2521, Am2946, Am2947, Am29520, Am29521, Am29827, Am29828, Am29841, Am29842, Am29843, Am29844, Am29845, Am29846, Am7303, Am7304, Am7307, Am7308, AmZ8121, Am81LS95, Am81LS96, Am81LS97, Am81LS98, Am8303, Am8304, Am8307, Am8308, Am93S10, Am93S16, Am9341, MC4015, MC6880, MC6882, MC6885, MC6886, MC6887, MC6888, MC6889, MC14495, DS1630, DS1631, DS1632, DS1633, DS1634, DS3630, DS3631, DS3632, DS3633, DS3634, DS3662, DS16149, DS16179
Misc Logic 2	BA12003, CA3045, CA3046, CA3081, CA3082, CA3083, CA3086, CA3161, L20x, L70x, QS3384, TD6208x, TD6278xRev8, ULN200x, ULN201x, ULN202x, ULN2064, ULN2074/76, ULN280x, ULN282x, UDN6118, UDN298xRev8, LMx24, LMx39, LM2900, uA741, FCJ121Rev1, D345, D346, D492, D718, K155IE1, KR559IP1, KR559IP2, SNG4xRev5, SNG6xRev5, SNG9xRev5, SNG13xRev5, SNG14xRev5, SNG15xRev5, SNG16xRev5, SNG19xRev5, SNG22xRev5, SNG23xRev5, SNG24xRev5, SNG26xRev5, SNG32xRev5, SP302A, SP304A, SP305A, SP306A, SP314A, SP317A, SP334A, SP337A, SP357A, SP358A, SP370A, SP374A, SP375A, SP377A, SP380A, SP381A, SP384A, SP387A, SP391A, CBM251641-02, CBM906114-01,

Zum Test der Ost-Typen kann auch das entsprechende West-Pendant gewählt werden.

Einige Tests wurden nach Datenblättern implementiert. Die Übersichtsliste für Logikchips gibt eine Übersicht, welche Tests verifiziert wurden. Sollte bei einem Logikchip ein „false negative“, also ein korrekter Chip als fehlerhaft, gemeldet werden, dieses bitte als Fehler mitteilen.

Logik-Chips	
PAL	Auslesen von PALs (nicht „registered“, nur rein kombinatorische Typen), z.B. PAL10H8, PAL12H6, PAL14H4, PAL16H2, PAL10L8, PAL12L6, PAL14L4, PAL16L2, PAL16C1

Programmierbare Chips	
EPROM	2708 (Adapter erforderlich)
EPROM	2716/2516 (Adapter erforderlich), auch Hitachi HN48016 EEPROM
EPROM	TMS2716 (Adapter erforderlich)
EPROM	2532 (Adapter erforderlich)
EPROM	2732 (Adapter erforderlich)
EPROM	2564 (Adapter erforderlich)

Weitere Features:

- Über 120 verschiedene SRAMs (>700 auf der Vergleichsliste), mehrere FIFO RAMs und über 30 verschiedene DRAMs (>250 auf der Vergleichsliste) können getestet werden.
- Es können zwölf SRAMs, drei DRAMs und drei ROMs selbst definiert werden.
- Über 80 verschiedene ROMs, PROMs und EPROMs können ausgelesen werden (>650 auf der Vergleichsliste).
- ROMs können über die CRC32 identifiziert werden (externe Datenbankdateien enthalten ca. 400.000 ROMs).
- Von über 420 EPROMs können Hersteller und Typ identifiziert werden.
- Sechs EPROMs können programmiert werden.
- Über 1200 Logik-ICs können geprüft werden.
- Einfacher Taktgenerator (300 Hz bis 8 MHz)
- Einfacher Frequenzzähler (1 Hz bis 2 MHz)
- Suche eines Chips über eine eingebaute Datenbank mit knapp 750 Vergleichstypen.
- Der Inhalt der unterstützten ROMs / PROMs / EPROMs kann auf einer SD-Speicherkarte gesichert werden (geeigneter SD-Kartenadapter vorausgesetzt).
- PALs/GALs können (mit Einschränkungen) kombinatorisch ausgelesen werden.
- Alle Pins besitzen einen Überspannungsschutz bis 40V und sind Kurzschlussfest.

Da der Tester einen recht günstigen ATmega2560 einsetzt, mussten einige kleinere Kompromisse eingegangen werden. Auf die bekannten Probleme und Einschränkungen wird in Abschnitt 7 näher eingegangen. Auch wenn nicht alle Fehler erkannt werden können, dürften für die meisten Anwendungsfälle die Ergebnisse jedoch ausreichend valide sein.

2 Bedienung

2.1 Tastenbelegung

Die Tasten sind teilweise mehrfach belegt.

Im IC-Menü / IC-Submenü:

SELECT BACK	JUMP FWRD	OK	FUNC TOP	Function
X				einen Chip zurück
	X			einen Chip weiter
X	X			Sprung zum Ausgang ¹
			X	Sprung zum Hauptmenü ² oder ins Untermenü
		X		Teste ausgewählten Chip
		gedrückt halten		alternativer Mode (in einigen Fällen), z.B. Speichern

Im TTL-Menü:

SELECT BACK	JUMP FWRD	OK	FUNC TOP	Function
X				einen Chip zurück
gedrückt halten				schnell zurück
	X			einen Chip weiter
	gedrückt halten			schnell weiter
X	X			Sprung zum Ausgang / Beende Loop-Modus
			X	Verlasse Menü / Beende Loop-Modus
		X		Teste ausgewählten Chip

¹ nur im Untermenü

² nur im Untermenü

Im Such-Menü:

SELECT BACK	JUMP FWRD	OK	FUNC TOP	Function
X				einen Eintrag weiter (gleicher Präfix)
	X			nächstes Zeichen
X	X			zurück zum Anfang
			X	zurück zum Anfang
		X		Auswahl

Im Config-Menü:

SELECT BACK	JUMP FWRD	OK	FUNC TOP	Function
X				einen Eintrag zurück
	X			einen Eintrag weiter
X	X			Sprung zum Ausgang
			X	Sprung zum Ausgang
		X		Auswahl

Im PAL-Menü:

SELECT BACK	JUMP FWRD	OK	FUNC TOP	Function
X				Cursor nach rechts
	X			Wechseln innerhalb der Gruppe [I] = Eingabe / [O] = Ausgabe / [-] = Ignoriert bzw. [G] = GND / [V] = Vss
X	X			Wechsel der Gruppe [IO-] und [GV]
			X	Wechsel der Gruppe [IO-] und [GV]
		X		Auswahl

TIPP:

Anstelle die (langen) Taster zu drücken, ist es einfacher diese leicht zur Seite zu kippen. Insbesondere zwei Taster können so sehr leicht gleichzeitig ausgelöst werden.



2.2 Das Menü

Das Hauptmenü ist inzwischen recht groß geworden. Aus diesem Grund wurden die Chips bzw. Funktionen in mehreren Untermenüs angeordnet. Im Hauptmenü kann ausgewählt werden:

Hauptmenü
SRAMs common
SRAMs uncommon
SRAMs cypress
DRAMs common
DRAMs uncommon
FIFO
EPROM
PROM/ROM 1
PROM/ROM 2
LOGIC
CUSTOM
PROGRAMMING
SERVICE

Abbildung 2.1: Hauptmenü

Das Menü gibt nur einen „Haupttypen“ vor, mit dem noch weitere ICs getestet werden können. Aus den Vergleichslisten geht hervor, welche Einstellung konkret für einen expliziten Chip korrekt ist.

Menu (long form)	Menu (short form)
SRAM (common)	
2102 - 1k x 1	2102
2115 - 1k x 1	2115
2114 - 1k x 4	2114
6550 - 1k x 4	6550
2142 - 1k x 4	2142
4118A - 1k x 8	4118A
2147 - 4k x 1	2147
xx16 - 2k x 8	xx16
xx64 - 8k x 8	xx64
xx256 - 32k x 8	xx256
xx512 - 64k x 8	xx512
xx1024 - 128k x 8	xx1024

Menu (long form)	Menu (short form)
SRAM (uncommon)	
3101 - 16 x 4	3101
74219 - 16 x 4	74219
4505 - 64 x 1	4505
74910 - 64 x 4	74910
93419 - 64 x 9	93419
6810 - 128 x 8	6810
3106 - 256 x 1	3106
93410 - 256 x 1	93410
2101 - 256 x 4	2101
2111 - 256 x 4	2111
2112 - 256 x 4	2112
74920 - 256 x 4	74920
74921 - 256 x 4	74921
6561 - 256 x 4	6561
81C50 - 256 x 8	81C50
81C51 - 256 x 8	81C51
81C54 - 512 x 8	81C54
82S208 - 256 x 8	82S208
82S210 - 256 x 9	82S210
82S212 - 256 x 9	82S212
8185 - 1k x 8	8185
74170/74670 - 4 x 4	74170/670
TC4039 - 4 x 8	4039
7481/7484 - 16 x 1	7481/84

Menu (long form)	Menu (short form)
SRAM (cypress)	
x150 - 1k x 4	x150
x167 - 16k x 1	x167
x168 - 4k x 4	x168
x1681 - 4k x 4	x1681
x170 - 4k x 4	x170
x187 - 64k x 1	x187
x188 - 16k x 4	x188
x198 - 16k x 4	x198
x198A - 16k x 4	x198A
x1981 - 16k x 4	x1981
x423 - 256 x 4	x423
x1026 - 256k x 4	x1026
x1257 - 256k x 1	x1257
x1258 - 64k x 4	x1258
x1281 - 64k x 4	x1281
x1298 - 64k x 4	x1298
x1299 - 64k x 4	x1299
xx257 - 32k x 8	xx257

Menu (long form)	Menu (short form)
DRAM (common)	
2104 - 4k x 1	2104
2107 - 4k x 1	2107
4116 - 16k x 1	4116
2118 - 16k x 1	2118
4416 - 16k x 4	4416
4164 - 64k x 1	4164
4464 - 64k x 4	4464
41256 - 256k x 1	41256
44256 - 256k x 4	44256
41024 - 1024k x 1	41024

Menu (long form)	Menu (short form)
DRAM (uncommon)	
3732L - 32k x 1	3732L
3732H - 32k x 1	3732H
4532-3 - 32k x 1	4532-3
4532-4 - 32k x 1	4532-4
2108L - 8k x 1	2108L
2108H - 8k x 1	2108H
4108-x0 - 8k x 1	4108-x0
4108-x1 - 8k x 1	4108-x1
5298A - 8k x 1	5298A
5298B - 8k x 1	5298B
4408NLT – 8k x 4	4408NLT
4408NLB – 8k x 4	4408NLB
4132 M1 - 32k x 1	4132 M1
4132 M2 - 32k x 1	4132 M2
4332 M1 - 32k x 1	4332 M1
4332 M2 - 32k x 1	4332 M2
41128 TOP - 64k x 1	41128 T
41128 BOT - 64k x 1	41128 B
SIMM30 - 256k x 8	SIMM
SIMM30 - 1M x 8	SIMM
SIMM30-P - 256k x 1	SIMM P.
SIMM30-P - 1M x 1	SIMM P.
SIMM30-P IBM - 256k x 1	SIMM P.
SIMM30-P IBM - 1M x 1	SIMM P.
ZIP20 - 64k x 4	ZIP64
ZIP16 - 256k x 1	ZIP256
ZIP20 - 256k x 4	ZIP256
ZIP20 - 1024k x 1	ZIP1024
ZIP20 - 1024k x 4	ZIP1024
4008 – 1k x 1	4008
6605 - 4k x 1	6605

Menu (long form)	Menu (short form)
FIFO RAMs	
40105 (16 x 4)	40105
74222/27 (16 x 4)	74222/27
74224/28/32 (16 x 4)	74224/28
74225 (16 x 5)	74225
74229/33 (16 x 5)	74229/33
74234/36 (64 x 4)	74234/36
74235 (64 x 5)	74256
747404 (64 x 5)	747404
7200 (256 x 9)	7200
7201 (512 x 9)	7201
7202 (1024 x 9)	7202
7203 (2048 x 9)	7203
7204 (4096 x 9)	7204
7200 (256 x 9) PLCC32	7200 PLCC
7201 (512 x 9) PLCC32	7201 PLCC
7202 (1024 x 9) PLCC32	7202 PLCC
7203 (2048 x 9) PLCC32	7203 PLCC
7204 (4096 x 9) PLCC32	7204 PLCC

Menu (long form)	Menu (short form)
EPROMs	
2716 - 2k x 8	2716
2732 - 4k x 8	2732
2764 - 8k x 8	2764
27128 - 16k x 8	27128
27256 - 32k x 8	27256
27512 - 64k x 8	27512
271001 - 128k x 8	271001
272001 - 256k x 8	272001
274001 - 512k x 8	274001
1702 - 256 x 8	1702
2704 - 512 x 8	2704
2708 - 1k x 8	2708
TMS2716 – 2k x 8	TMS2716
TMS2564 – 8k x 8	TMS2564
VCS/2600 Cart	VCS/2600
ES 2764-27128	2764+
ES 27256-27512	27256+

Menu (long form)	Menu (short form)
ROMs / PROMs 1	
2308 - 1k x 8	2308
2316A - 2k x 8	2316A
2316B - 2k x 8	2316B
2332 - 4k x 8	2332
2364(24) - 8k x 8	2364(24)
2364(28) - 8k x 8	2364(28)
23128 - 16k x 8	23128
23256 - 32k x 8	23256
23512 - 64k x 8	23512
231000 - 128k x 8	231000
232000 - 256k x 8	232000
234000 - 512k x 8	234000
6540 - 2k x 8	6540
RO-3-2513 – 64x8x5	2513
6670 – 128x7x5	6670
2804 – 512 x 8 EEPROM	2804

Menu (long form)	Menu (short form)
ROMs / PROMs 2	
7488/188/288 - 32x8	7488
74186 - 64x8	74186
4187/287/387 - 256x4	74187
74S271/470 - 256x8	74S271
74S270/570 - 512x4	74S270
74S472/473 - 512x8	74S472
74S474/475 - 512x8	74S474
74S476/572 – 1k x4	74S476
74S478/479 – 1k x8	74S478
82S184/185 – 2k x4	82S184
82S190/191 – 2k x8	82S190
82S195 – 4k x4	82S195
82S321 – 4k x8	82S321
82S641 – 8k x8	82S641

Menu (long form)	Menu (short form)
LOGIC	
7400+ Logic Tests	
74300+ Logic Tests	
74600+ Logic Tests	
75000+ Logic Tests	
4000 Logic Tests	
4500 Logic Tests	
Misc Logic Tests 1	
Misc Logic Tests 2	
USSR TTL Logic	
USSR CMOS Logic	
Misc Tests	
PAL 20pin	PAL
PAL 24pin	PAL
Identify Logic IC	
Joystick Test	
Visual Fuzzy Test	
Clock Generator	
Frequency Counter	

Menu (long form)	Menu (short form)
CUSTOM SRAMs / DRAMs	
12 x SRAMs	
3 x DRAMs	
3 x ROMs	

Menu (long form)	Menu (short form)
PROGRAMMING	
2708	
2716 / 2516	
TMS2716	
2532	
2732	
2564	

Menu (long form)	Menu (short form)
SERVICE	
Search	
Info	
Enter Config	
Selftest	

„Gelb“ hinterlegte Einträge benötigen einen speziellen Adapter.

Übersicht über die Aufteilung des „LOGIC“ Menüs:**7400+ Logic**

7400 01 74H01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 30
31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 74LS51 52 53 74H53 54 74H54
74LS54 74H55 74LS55 56(*) 57 58 60 61(*) 62 64 65 68(**) 69(**) 70 74H71 74L71 72 73 74 75 76
74LS76 77 74H78 74L78 74LS78 79(*) 80 82 83 85 74C85 86 74L86 87 90 74C90 91 92 93 74C93
74L93 94 74C95 74L95 95 96 97 98 99 100 101 102 103 104 105 106 107 74LS107 108 109 110 111
112 113 114 115 116 118 119 120 121 122 123 125 126 128 130 131 132 133 134 135 136 137 138
139 140 141 142 143 144 145 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
163 164 165 166 167 168 169 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 190
191 192 193 194 195 196 197 198 199 221 230(*) 231 237 238 240 241 242 243 244 245 246 247 248
249 251 253 255(**) 256 257 258 259 260 261 265 266 269 273 275 276 278 279 280 283 284 285 286
290 292 293 294 295 298 299

74300+ Logic

74322 323 340 341 344 347 348 350 351(**) 352 353 354 355 356 357 363 364 365 366 367 368 373
374 375 376 377 378 379 381 74F381 382 74F382 384 385(*) 386 388(**) 390 393 395 396 398 399
74S405 412 74S416 422 423 425 426 74F432(**) 436(**) 437(**) 440 441 442 443 444 445 446 447
448 449(**) 456 461 465 466 467 468 490 518 519 520 521 522 524 526 527 528(*) 533 534 535 536
537 538 539 540 541 543 544 545 546(**) 547(**) 74F547 74F548(**) 560 561 563 564 566(**) 567(**)
568 569 74LS573 574 575 576 577 579 580 583 588 589 590 591(*) 592 593 594 595 596 597 598 599

74600+ Logic

74604 605 606 607 620 621 622 623 638 639 640 641 642 643 644 645 646 647 648 649 651 652 653
654 657 666 667(*) 668 669 671 672(*) 673 674 677 678 679 680 681 682 683 684 685 686 687 688
689 690 691 692 693 696 697 698 699 720(*) 721 740(*) 741(**) 744(*) 748 756 757 758 759 760 762
763 779 793(*) 794 795 796 797 798 800 802 804 805 808 810 821 822 823 824 825 826(*) 827 828
832 841 842 843 844 845 846 848 850(**) 851(**) 857(**) 861(*) 862(*) 863(*) 864(*) 866 867 869 873
874 876 878 879(**) 880 881(*) 885 74ALS900 74C901 74ALS902 74C902 74ALS903 74C903 74C904
74C906 74C914 74C915 74S940(*) 74S941(*) 990 991 992(*) 993 994(*) 995 996 1000 1002 1003
1004 1005 1008 1010 1011 1020 1032 1034 1035 1240 1241 1242 1243 1244 1245 1620 1621 1622
1623 1638 1639 1640 1641 1642 1643 1644 1645 2540 2541 3037 7001 7002(*) 7014(*) 7032(*) 7266
8541(*) 9034(*) 9035(*) 9114(*) 9115(*) 9134(*) 9135(*) 9240(*) 9244(*) 9245(*)

75000+ Logic

75121(**) 122(**) 123 124 125(**) 127 138 140 154 160 172 173 189 361-8 361-14 450 451 452 453
454 460 461 462 463 464 466 467 468 469 470 471 472 473 474 491 492 494 497 498

4000+ Logic

4000 01 02 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
35 38 40 41 42 43 44 48 49 50 51 52 53 54 55 56 59 60 63 66 67 68 69 70 71 72 73 75 76 77 78 81 82
85 86 93 94 95 96 97 98 99 014 032 097 098 101 102 103 104 106 107 109 110 116 147 160 161 162
163 174 175 192 193 194 240 244 245(*) 257(*) 373(*) 374 511 4316 4428

4500+ Logic

4501 02 03 04 06 07 08 10 11 12 13 14 15 16 18 19 20 22 26 28 29 30 31 32 38 39 43 51 55 56 58 60
61 72 74 84 85 98 4723(**) 4724 4929 4930 4931 4934(**) 4935(**) 49700 49701 49702 49703
49704(**) 49705 49710(**) 49711 49713(**) 49714(**) 49805(*)

Misc Logic 1

3205 3212 3216 3226 3404 6529 7640(*) 7641 7707 7708 7709 7711 7712 7713 7714 7715 7810(*)
7811(*) 7812(*) 7819(*) 7833 7835 7837 7838 8708 8713 7093(*) 7094(*) 7131 7160 7220 74L90 74L93
8090 8091 8092 8093(*) 8094(*) 8095(*) 80C95(*) 8096(*) 80C96(*) 8097(*) 80C97(*) 8098 80C98 8121
8123 8131 8160 8200 8205 8212 8216 8219 82C19 8220 8226 8230(*) 8231(*) 8233(*) 8234(*) 8235(*)
8241 8242 8250(*) 8251(*) 8252 82S62 8263 8264 8266 8267 8270 8271 8280 8281 8282 8283 8286
8287 8290(*) 8291(*) 8292(*) 8293(*) 8415A(*) 8455A(*) 8470A(*) 8471A(*) 8480A(*) 8481A(*) 8490A(*)
8530 8532(*) 8533(*) 8555 8556 8560 8563 8570(*) 8590(*) 8640(*) 8641 86L75 86L76 86L93 8808A(*)
8810(*) 8811(*) 8812(*) 8815A 8816A(*) 8819(*) 8829A(*) 8830 8833 8835 8835 8837 8838 8840A(*)
8848A(*) 8855A(*) 8870A(*) 8875A(*) 8880A(*) 8881A 8885A(*) 8890A(*) 8891A(*) 9000(*) 9001(*)
9002(*) 9003(*) 9004(*) 9007(*) 9009 9012(*) 9016(*) 9017(*) 9024(*) 9300(*) 9301(*) 9308 9309 9310
9312(**) 9314 9316 9317 9318 9321(*) 9322 9324(**) 9334 93S47 93S62 9368 9370 9602 8H16(*)
8H70(*) 8H80(*) 8H90(*) 8T09(*) 8T10 8T13 8T14 8T22(*) 8T23 8T24 8T26 8T28 8T37 8T38 8T80(*)
8T90(*) 8T93(*) 8T94(*) 8T95 8T96 8T97 8T98 8T125(*) 8T245 9H50 9H51 Am25LS07(*) Am25LS08(*)
Am25LS09 Am25S10 Am25LS14 Am25LS15(**) Am25S18 Am25LS22(*) Am25LS23(*)
Am25LS2518(**) Am25LS2519(**) Am25LS2521 Am2946 Am2947 Am29520A(*) Am29521A Am29827
Am29828 Am29841 Am29842 Am29843(*) Am29844(*) Am29845(*) Am29846(*) Am7303 Am7304
Am7307 Am7308 AmZ8121 Am81LS95 Am81LS96 Am81LS97 Am81LS98 Am8303 Am8304 Am8307
Am8308 Am93S10 Am93S16 Am9341 MC3482 MC4015 MC6880 MC6882 MC6885 MC6886 MC6887
MC6888 MC6889 MC14495 DS1630(*) DS1631 DS1632 DS1633(*) DS1634(*) DS3630(*) DS3631
DS3632 DS3633 DS3634 DS3662 DS16149 DS16179 DS36149 DS36179

Misc Logic 2

BA12003 CA3045(**) CA3046 CA3081 CA3082 CA3083 CA3086 CA3161 L20x L70x QS3384 TD6208x
TD6278x ULN200x ULN201x ULN202x ULN2064 ULN2074 ULN280x ULN282x UDN6118 UDN298x
LMx24 LMx39 LM2900 LM3900 uA741 FCJ121 D345 D346 D492 D718 K155IE1 KR559IP1 KR559IP2
SNG40(*) SNG60(*) SNG90(*) SNG130(*) SNG140(*) SNG150(*) SNG160(*) SNG190(*) SNG220(*)
SNG230 SNG240(*) SNG260(*) SP302A(*) SP304A(*) SP305A(*) SP306A(*) SP314A SP317A(*)
SP334A(*) SP337A(*) SP357A(*) SP358A(*) SP370A(*) SP374A(*) SP375A(*) SP377A(*) SP380A
SP381A(*) SP384A(*) SP387A(*) SP391A(*) CBM251641-02 CBM906114-01

Misc Tests

DIS1417 DL1414 DL2416 HP730x HP7340 TIL306/307 TIL308/309 TIL311 VQB76

(*) getestet, nicht verifiziert, (**) ungetestet

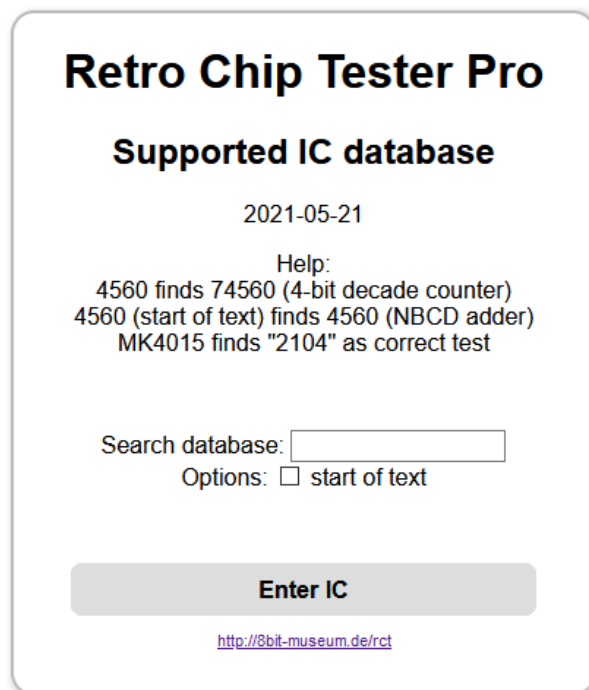
2.3 Suchen eines ICs / Unterstützte ICs

Das Menü zeigt nur einen IC stellvertretend für mehrere ICs desselben Typs an. Zum Ermitteln des korrekten Tests werden vier PDF-Dateien und eine HTML-Datei zur Verfügung gestellt.

Die PDF-Dateien enthalten eine Übersicht der unterstützten SRAMs, DRAMs, (EP)ROMs und Logik-ICs und welcher Test für den jeweiligen Baustein verwendet werden kann.

Die HTML-Datei enthält dieselben Informationen zusammengefasst mit einer Abfragemaske. Da einige ICs gleiche Bezeichnungen verwenden und die HTML-Datei nur den jeweils ersten gefundenen IC anzeigt, kann diese die PDF-Dateien jedoch nicht vollständig ersetzen.

Nach dem Aufruf erscheint folgende Webseite:



The screenshot shows a web interface titled "Retro Chip Tester Pro" with a subtitle "Supported IC database". Below the title is the date "2021-05-21". A "Help:" section contains three lines of text: "4560 finds 74560 (4-bit decade counter)", "4560 (start of text) finds 4560 (NBCD adder)", and "MK4015 finds '2104' as correct test". There is a search input field labeled "Search database:" and a checkbox labeled "Options: ☐ start of text". At the bottom is a large grey button labeled "Enter IC" and a URL link "[http://8bit-museum.de/rct](\"http://8bit-museum.de/rct\")".

Abbildung 2.2: Startbildschirm

Nach der Eingabe einer IC-Bezeichnung wird die notwendige Testeinstellung hinter „Select:“ ausgegeben. Bei einem „MK4015“ wäre dieses der Test „2104“.

Retro Chip Tester Pro

Supported IC database

2021-05-21

Help:
4560 finds 74560 (4-bit decade counter)
4560 (start of text) finds 4560 (NBCD adder)
MK4015 finds "2104" as correct test

Search database:

Options: ☐ start of text

Mostek MK4015

DRAM (4k x 1)

Status: supported

Select: 2104

Alternatives:

D2104A, FM4027, MB8227, IM7027,
ITT4027, MK4027, MK4096, MK4200,
MCM4027, MCM6604, uPD414, 2660,
4027, TMS4027

<http://8bit-museum.de/rct>

Abbildung 2.3: Suche nach MK4015

Achtung: Die „Alternativen“ können alle mit diesem Test geprüft werden, sind aber nicht unbedingt als Ersatz im realen System geeignet.

iPhone/iPad User:

Die Datei nicht aus einer E-Mail heraus öffnen (Javascript wird dann nicht ausgeführt), sondern auf dem Gerät ablegen (oder in einer Dropbox, Google Drive etc.) und dann öffnen.

3 Testen von Chips

3.1 Testen von SRAMs und DRAMs

3.1.1 Einsetzen von Chips

Alle Chips werden linksbündig eingesetzt. Es gibt ein paar wenige Ausnahmen, bitte die Hinweise bei den TTL Chips beachten.

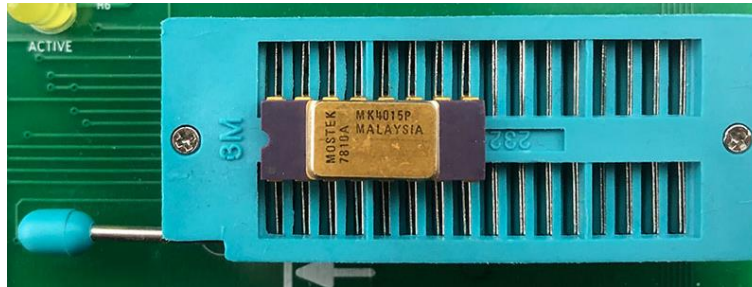


Abbildung 3.1: Korrekte Position des Chips

Sollte hinter der Typenbezeichnung ein „Pos x“ ausgegeben werden, so muss der Chip an „Position x“ in den Sockel eingesetzt werden. Normalerweise werden die Chips linksbündig eingesetzt (entspricht „Pos 1“). Da einige Chips die Spannungsversorgung an nicht an den üblichen Pins (z.B. 7475, 7490, 7492, 7493) erwarten, ist dieser Chip dann versetzt einzusetzen („Pos 4“ bedeutet also, dass drei Reihen frei bleiben müssen).



Abbildung 3.2: Chip eingesetzt an Position 4 (7490)

Bevor der eigentliche Test beginnt, führt der RCT einen Vortest durch, um zu prüfen, ob der richtige IC eingesetzt wurde bzw. der Chip überhaupt getestet werden kann. Wenn der Vortest fehlschlägt, wird sofort die Meldung „<IC type> Test failed“ ausgegeben. Der Vortest kann in der Konfiguration ausgeschaltet werden („Chip detection“).



3.1.2 Auswahl des korrekten Bausteins

Das Menü erlaubt die Auswahl des zu testenden Chips. Aufgrund des begrenzten Platzes wird nur ein einzelner „Haupttyp“ angezeigt. Die Kompatibilitätsliste für SRAMs / DRAMs gibt Auskunft, welcher Menüpunkt im Einzelfall korrekt ist (Beispiel: Ein Am2148 ist identisch mit einem 2114 (1k x 4), d.h. Menüauswahl „2114“).



Abbildung 3.3: Beispiele: Auswahl eines „2114“ SRAM bzw. „4116“ DRAM

Sollte der Typ nicht aus der Bezeichnung direkt ablesbar sein, kann von der Website je eine Vergleichsliste für SRAMs, DRAMs und ROMs heruntergeladen werden, welche die korrekte Einstellung für viele Speicherbausteine auflistet.

3.1.3 Ausgabe des Testergebnis

Bei einem erfolgreichen Test wird `passed` ausgegeben. Schlägt der Test fehl, wird die defekte Speicherzelle (z.B. A:13F4) und der erwartete und gelesene Wert ausgegeben (z.B. 5>F = Wert 1010 erwartet, gelesen wurde 1111).

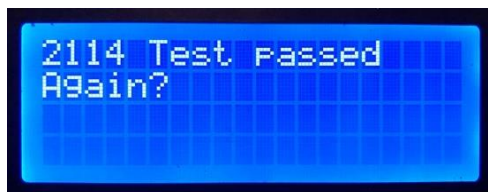


Abbildung 3.4: Erfolgreicher Test eines 2114 SRAM



Abbildung 3.5: Fehlgeschlagene Tests eines 6550 (erwartet b0000, gelesen b0010) an Adresse 0x02C0 und 4116 (erwartet 1, gelesen 0) an Adresse 0x3840

Der Testablauf ist wie folgt:

#	Test	Beschreibung	Anzeige bei Fehler / Grund
1	Pre-Test	Es wird geprüft, ob der Speicherbaustein grundsätzlich ansprechbar ist. Test wird durchgeführt, wenn in der Konfiguration „Chip detect:“ auf „1“ (Standard) oder „2“ gesetzt ist.	„Pre-Test“ - kein IC eingesetzt - IC defekt
		Test, ob „Chip Enable“ funktionsfähig. Test wird bei SRAM durchgeführt, wenn in der Konfiguration „Chip detect:“ auf „2“ gesetzt ist.	„Defect CE“ „Chip Enable“ Signal defekt
		Test, ob Adressdekoder funktionsfähig. Test wird durchgeführt, wenn in der Konfiguration „Chip detect:“ auf „2“ gesetzt ist.	„AddrDecoder: HHHHHH“ - Adressdekoder defekt - evtl. Speicherfehler
2	Haupttest	Die Speichermatrix wird getestet.	„A: HHHHHH H>H“ Speicherfehler

Hinweis:

Standardmäßig wird der „Pre-Test“ durchgeführt (in der Konfiguration ist „Chip detect: 1“ standardmäßig eingestellt). Mit „Chip detect: 2“ werden erweiterte Tests aktiviert (CE und Adressdekoder). Wird „Chip detect: 0“ eingestellt, sind die „Pre-Tests“ vollständig deaktiviert. Die „Pre-Tests“ werden auch übersprungen, wenn die Testergebnisse gespeichert werden sollen (siehe Kapitel 3.1.4).

Fehler im Adressdekoder

Bei einem Fehler im Adressdecoder wird ein Hexadezimalwert angezeigt. Jedem Bit in diesem Wert ist eine Adressleitung zugeordnet. Wenn Bit 0 gesetzt ist, ist A0 fehlerhaft, wenn Bit 1 gesetzt ist, dann A1 usw. Bei DRAM wird in den unteren Bits die Zeilenadressen und in den oberen Bits die Spaltenadressen ausgegeben.

Beispiel:

SRAM mit Ausgabe "AddrDecoder: 010080" (0000 0001 0000 0000 1000 0000):

Bit 7 und Bit 16 sind gesetzt → A7 und A16 sind fehlerhaft

DRAM (4116) mit Ausgabe "AddrDecoder: 000102" (0000 0000 0000 0001 0000 0010):

Ein 4116 besitzt sieben Adressleitungen für Zeilen und genauso viele für Spalten, daher Bit 0-6 für Zeilen/RAS, Bit 7–13 für Spalten/CAS) → Bit 1 ist gesetzt = A1 mit RAS, Bit 8 ist gesetzt = A1 mit CAS fehlerhaft.

Wichtig:

Der Chip Tester erkennt defekte Speicherzellen sehr zuverlässig. Wird derselbe Speicherbaustein mehrfach getestet und der Fehler tritt immer an derselben Speicherzelle auf, dann ist diese mit großer Sicherheit defekt und damit der Speicherbaustein.



Leider gibt es auch Fehler, die nicht so eindeutig sind:

Wird ein Speicherbaustein mehrfach getestet und tritt der Fehler an unterschiedlichen Speicherzellen auf, dann ist es durchaus möglich, dass der Speicher zu langsam für den Test ist (d.h. die Zugriffszeit über die Jahre größer geworden ist), die Spannungspegel am Ausgang nicht mehr der Spezifikation entsprechen oder die Eingänge einen höheren Strom benötigen, den der Tester nicht zur Verfügung stellen kann (max. 8mA). Das Kapitel 7 geht auf weitere mögliche Probleme ein.

Diese Speicherbausteine können in einem Heimcomputer noch funktionieren. Erhöhte Zugriffszeiten fallen i.d.R. nicht auf. Auch bei Signalpegeln, die nicht mehr genau der Spezifikation entsprechen (genau können diese nicht eingehalten werden, siehe auch Kapitel 7), kann ein Computer noch fehlerfrei funktionieren (zumindest eine gewisse Zeit lang). Der Tester wird aber bei langsamen Bausteinen oder einem unzureichenden Signalpegel einen Fehler melden.

Es wird nicht getestet, ob ein Baustein in einem bestimmten Computer noch funktioniert, sondern ob die vorgegebenen Grenzen eingehalten werden.

Die Testdauer beträgt ungefähr:

8k x 8 SRAM (6264)	
Pattern Tests (0-Test, 1-Test, etc.)	ca. 5s je Test
Random Test	ca. 7s
March Y	ca. 20s
March U	ca. 40s

64k x 8 SRAM (62512)	
Pattern Tests (0-Test, 1-Test, etc.)	ca. 30s je Test
Random Test	ca. 55s
March Y	ca. 2:00min
March U	ca. 3:30min

16k x 1 DRAM (4116)	
Pattern Tests (0-Test, 1-Test, etc.)	ca. 1,5s je Test
Random Test	ca. 4,5s
March Y	ca. 8s
March U	ca. 13s

1024k x 8 SIMM/SIPP	
Pattern Tests (0-Test, 1-Test, etc.)	ca. 1:30min je Test
Random Test	ca. 5:00min
March Y	ca. 11:30min
March U	ca. 21:00min

3.1.4 Speichern des Testergebnisses

Die bei einem Test gelesenen Inhalte können auf einer SD-Speicherkarte gesichert werden. Hierzu muss

- ein Kartenadapter mit einer geeigneten Speicherkarte angeschlossen sein,
- die **SD-Kartenunterstützung** in der Konfiguration eingeschaltet sein („**SD Card: 1**“),
- die **Dump-Funktion** in der Konfiguration eingeschaltet sein („**Alt. mode: 1**“).



Wird der Speichertest mit einem **langem Druck auf die OK-Taste** gestartet, wird die Datei `<chip>.<xxx>` erzeugt, wobei „xxx“ eine fortlaufende Zahl ist. Es können also mehrere gleiche Chiptypen nacheinander getestet werden.

Der Testvorgang wird nicht – wie bei einem normalen Test - bei einem Fehler direkt abgebrochen, sondern es werden alle Tests durchgeführt und erst am Ende wird der erste gefundene Fehler angezeigt.

Der Datei enthält den gesamten Chipinhalt von allen Tests hintereinander. Bei einem 6550 SRAM (1 kByte x 4) ist die Datei wie folgt aufgebaut:

0x0000	0-Test (0000)
0x0400	5-Test (0101)
0x0800	A-Test (1010)
0x0C00	F-Test (1111)
0x1000	count+ (0,1,2,3,4...)
0x1400	count- (f, e, d, c, ...)
0x1800	Random(16) gelesen
0x1C00	Random(16) geschrieben

Bei einem 1-bittigen Speicher, entfallen ein paar Tests, somit ergibt sich folgende Aufteilung: 0-Test (0), 1-Test (1), 01-Test (0101), 10-Test (1010), Random(2).

Bei einem 8-bittigen Speicher werden dann 0x55, 0xAA, 0xFF beim Pattern-Test und Random(256) verwendet.

Beispiele für ein korrektes 6550 SRAM:

```
00000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

0-Test (0000)

```
00000400 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
00000410 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
00000420 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
00000430 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
```

5-Test (0101)

```
00000800 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a
00000810 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a
00000820 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a
00000830 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a 0a
```

A-Test (1010)

```
00000c00 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f
00000c10 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f
00000c20 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f
00000c30 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f 0f
```

F-Test (1111)

```
00001000 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
00001010 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
00001020 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
00001030 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
```

count+

```
00001400 0f 0e 0d 0c 0b 0a 09 08 07 06 05 04 03 02 01 00
00001410 0f 0e 0d 0c 0b 0a 09 08 07 06 05 04 03 02 01 00
00001420 0f 0e 0d 0c 0b 0a 09 08 07 06 05 04 03 02 01 00
00001430 0f 0e 0d 0c 0b 0a 09 08 07 06 05 04 03 02 01 00
```

count-

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00001800	0c	09	05	08	05	0d	0a	0a	05	01	0d	0b	0c	0a	06	05
00001810	0c	03	01	02	05	0e	00	0d	0f	0b	0e	01	03	0c	0d	0e
00001820	0b	06	06	0f	09	05	0b	04	03	08	0c	0d	08	0d	06	06
00001830	0c	0a	0a	07	02	01	05	00	0b	04	06	08	00	02	0a	0e
00001840	0d	00	0c	08	0c	02	0e	0d	0c	00	00	0e	0c	0f	08	03
00001850	0d	0e	03	0e	09	02	0e	05	06	04	03	05	01	09	00	06
00001860	05	0d	0b	0d	07	03	00	02	07	03	00	0a	05	08	06	0a

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00001c00	0c	09	05	08	05	0d	0a	0a	05	01	0d	0b	0c	0a	06	05
00001c10	0c	03	01	02	05	0e	00	0d	0f	0b	0e	01	03	0c	0d	0e
00001c20	0b	06	06	0f	09	05	0b	04	03	08	0c	0d	08	0d	06	06
00001c30	0c	0a	0a	07	02	01	05	00	0b	04	06	08	00	02	0a	0e
00001c40	0d	00	0c	08	0c	02	0e	0d	0c	00	00	0e	0c	0f	08	03
00001c50	0d	0e	03	0e	09	02	0e	05	06	04	03	05	01	09	00	06
00001c60	05	0d	0b	0d	07	03	00	02	07	03	00	0a	05	08	06	0a

Random (16) gelesen

Random (16) geschrieben

Beispiele für defekte 6550 SRAMs:

00000800	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0a	0a
00000810	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a
00000820	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a
00000830	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a	0a

Fehler im A-Test

00001000	01	01	03	03	05	05	07	07	09	09	0b	0b	0d	0d	0f	0f
00001010	01	01	03	03	05	05	07	07	09	09	0b	0b	0d	0d	0f	0f
00001020	01	01	03	03	05	05	07	07	09	09	0b	0b	0d	0d	0f	0f
00001030	01	01	03	03	05	05	07	07	09	09	0b	0b	0d	0d	0f	0f
00001040	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00001050	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f

Fehler im count+ Test

000000f0	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b
00000100	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b
00000110	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b	0b
00000120	0b	0b	0b	0b	0b	09	0b	0b	0b	0b	0b	0b	0b	09	09	09
00000130	09	09	09	09	09	09	09	09	0b	09	01	09	09	09	09	09
00000140	01	01	09	09	01	01	01	01	01	01	01	09	01	01	01	01
00000150	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
00000160	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
00000170	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
00000180	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
00000190	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
000001a0	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
000001b0	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
000001c0	01	01	01	01	01	01	01	01	01	01	01	01	01	00	01	01
000001d0	01	01	00	00	00	00	00	00	01	01	00	00	00	00	00	01
000001e0	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00	00
000001f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Fehler im 0-Test

Beispiele für ein defektes 2112A SRAM:

Es wurden zufällige Daten in den Speicher geschrieben, aber jeweils 16 aufeinanderfolgende Bytes sind identisch. Das deutet darauf hin, dass die Adressleitung A4 intern beschädigt ist, d.h. immer LOW ist. Die vorherigen Tests konnten diesen Fehler nicht erkennen (auch nicht der Count-Test, da der Speicher eine Breite von 4-Bit hat und das Zählmuster sich damit nach 16 Byte wiederholt).

```

00000700 08 0a 0c 0a 02 01 05 09 07 0d 0e 02 0b 08 0c 02
00000710 01 07 09 0e 01 0b 08 09 00 08 08 00 0f 08 00 0d
00000720 05 02 09 07 05 05 01 0e 0e 03 05 05 00 09 0b 06
00000730 04 0f 0a 07 09 01 07 0e 04 02 01 0e 0e 05 06 05
00000740 07 00 05 03 00 0e 07 05 05 0e 04 0a 06 0c 03 03
00000750 06 0f 0d 0f 02 0b 08 06 05 0e 05 05 05 08 0a 06
00000760 0c 00 0b 01 0c 0f 0d 0b 0a 02 06 05 06 08 02 01
00000770 08 02 01 02 0f 04 01 0b 00 09 05 0a 01 0e 0d 03
00000780 0a 0b 0f 0b 09 05 09 08 0f 0e 0f 0a 08 04 0e 05
00000790 09 07 02 09 0b 08 05 07 08 0d 00 0d 0a 0b 01 0f
000007a0 06 0f 07 03 04 0a 09 0c 02 04 00 0c 03 08 00 06
000007b0 02 0c 07 0b 0b 06 03 07 07 00 0d 0b 08 0b 0d 0c

```

Geschriebene Daten im Random(16) Test

```

00000600 01 07 09 0e 01 0b 08 09 00 08 08 00 0f 08 00 0d
00000610 01 07 09 0e 01 0b 08 09 00 08 08 00 0f 08 00 0d
00000620 04 0f 0a 07 09 01 07 0e 04 02 01 0e 0e 05 06 05
00000630 04 0f 0a 07 09 01 07 0e 04 02 01 0e 0e 05 06 05
00000640 06 0f 0d 0f 02 0b 08 06 05 0e 05 05 05 08 0a 06
00000650 06 0f 0d 0f 02 0b 08 06 05 0e 05 05 05 08 0a 06
00000660 08 02 01 02 0f 04 01 0b 00 09 05 0a 01 0e 0d 03
00000670 08 02 01 02 0f 04 01 0b 00 09 05 0a 01 0e 0d 03
00000680 09 07 02 09 0b 08 05 07 08 0d 00 0d 0a 0b 01 0f
00000690 09 07 02 09 0b 08 05 07 08 0d 00 0d 0a 0b 01 0f
000006a0 02 0c 07 0b 0b 06 03 07 07 00 0d 0b 08 0b 0d 0c
000006b0 02 0c 07 0b 0b 06 03 07 07 00 0d 0b 08 0b 0d 0c

```

Gelesene Daten im Random(16) Test

Hinweise:

Falls normalerweise durch den langen Druck auf die OK Taste eine alternative Funktion aufrufbar sein sollte, so ist diese solange nicht erreichbar, wie die Dump-Funktion in der Konfiguration eingeschaltet ist.



Durch die Verwendung der Dump-Funktion verändert sich das Timing beim Auslesen des Chips (dieser wird langsamer ausgelesen durch die notwendigen Zugriffe auf die SD-Speicherkarte). Dadurch kann sich das Ergebnis von dem Ergebnis ohne Dump-Funktion unterscheiden.



Bei DRAMs kann das Ergebnis verfälscht sein, wenn die Zugriffe auf die SD-Speicherkarte zu langsam stattfinden und dadurch ein Refresh des DRAMs nicht mehr gewährleistet ist.



Wenn die Daten gespeichert werden, wird ein evtl. aktivierter March-Y Test nicht durchgeführt.

3.1.5 Identifizieren von Page Mode, Nibble Mode und Static Column Mode bei DRAMs

Wenn in der Konfiguration die Identifizierung der DRAM Features eingeschaltet ist, versucht der Tester zu erkennen, ob der Speicherbaustein den Page Mode (PM), Nibble Mode (NM) und ggf. Static Column Mode (SC) unterstützt.

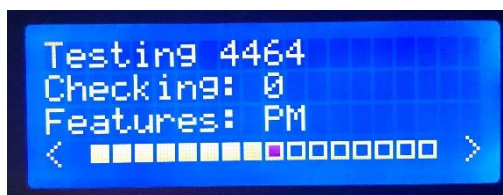


Abbildung 3.6: Speicher mit Page Mode

3.2 Spezielle Speicherbausteine

3.2.1 Testen von SIMM und SIPP Modulen

Es ist möglich 256 kByte und 1 Mbyte SIMM und SIPP Module zu testen.

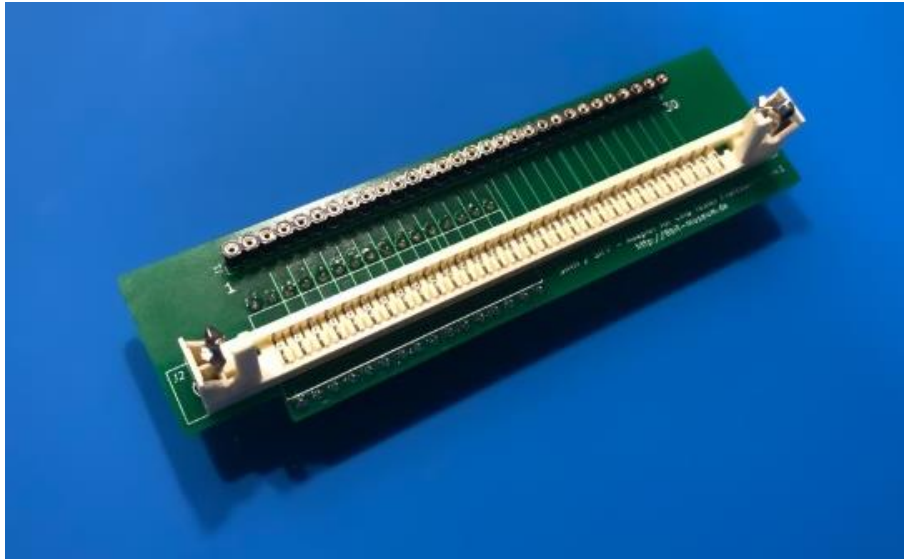


Abbildung 3.7: Adapter für SIMM und SIPP



Abbildung 3.8: Adapter im Sockel

Der Speicher wird in den Adapter gesteckt (entweder als SIMM oder als SIPP) und der Adapter linksbündig in den Sockel eingesetzt. Es gibt für jede Speichergröße zwei Tests, z.B. „SIMM 1024k x 8“ und „SIMM-P 1024k x 1“. Der Grund dafür ist, dass diese Module als 8-bittige und 9-bittige Module hergestellt wurden (also mit einem Parity-Bit). „SIMM NNN x 8“ testet also das eigentliche Modul, „SIMM-P NNN x 1“ das evtl. vorhandene Parity-Bit.

Normalerweise erkennt man Module mit Parity-Bit an der Bestückung mit drei oder mit neun Chips. Um die Kosten für Speichermodule zu senken, hatten einige Unternehmen Emulationslogik für das Paritätsbit für 30-Pin-SIMMs entwickelt. Diese Chips, wie der GS81C4100J70 oder der BP41C1000A-6, verwenden Logik zur Emulation des Paritätsbits, so dass kein echter Speicher vorhanden ist. In diesem Fall kann der Hauptspeicher („x8“) getestet werden, aber das Paritätsbit kann der RCT nicht testen, da kein Speicher zum Testen vorhanden ist.



Abbildung 3.9: SIMM mit Emulation des Paritätsbits

Ohne Adapter, können die Module auch anderweitig geeignet mit dem Tester verbunden werden.



Abbildung 3.10: SIPP-Adapter mit Breadboard

Der Sockel ist hierbei wie folgt belegt:

GND	52	1	32	21	Vcc
A0	51	2	31	22	D0
A1	50	3	30	23	D1
A2	49	4	29	24	D2
A3	48	5	28	25	D3
A4	47	6	27	26	D4
A5	46	7	26	27	D5
A6	45	8	25	28	D6
A7	44	9	24	29	D7
A8	43	10	23	30	/WR
A9	42	11	22	31	/RAS
A10	41	12	21	32	/CAS
A11	40	13	20	33	/CASP
	39	14	19	34	DP (in)
	38	15	18	35	QP (out)
	37	16	17	36	

Abbildung 3.11: Sockelbelegung bei SIMM/SIPP-Test

Ein 30-poliges SIMM bzw. SIPP Modul ist wie folgt belegt:

Pin	Name	Signal
1	VCC	+5 VDC
2	/CAS	Column Address Strobe
3	DQ0	Data 0
4	A0	Address 0
5	A1	Address 1
6	DQ1	Data 1
7	A2	Address 2
8	A3	Address 3
9	VSS	Ground
10	DQ2	Data 2
11	A4	Address 4
12	A5	Address 5
13	DQ3	Data 3
14	A6	Address 6
15	A7	Address 7
16	DQ4	Data 4
17	A8	Address 8
18	A9	Address 9
19	A10	Address 10
20	DQ5	Data 5
21	/WE	Write Enable
22	VSS	Ground
23	DQ6	Data 6
24	A11	Address 11
25	DQ7	Data 7
26	QP*	Data parity out
27	/RAS	Row Address Strobe
28	/CASP*	Parity Column Address Strobe
29	DP*	Data parity in
30	VCC	+5 VDC

Abbildung 3.12: SIMM/SIPP Pinbelegung

Für 256 kByte Module müssen die Adressleitungen A0 bis A8, die Datenleitungen D0-D7 sowie /CAS, /RAS und /WE (bzw. /WR) miteinander verbunden werden.

Für 1 MByte Module müssen die Adressleitungen A0 bis A9, die Datenleitungen D0-D7 sowie /CAS, /RAS und /WE (bzw. /WR) miteinander verbunden werden.

Weiterhin sind Vcc mit 5V und GND mit Vss zu verbinden.

Es können ausschließlich 5V Module getestet werden.



IBM SIMMs

In folgenden Karten und Geräten wurden spezielle IBM SIMMs eingesetzt:

- “SCSI Controller with Cache” (auch bekannt unter dem Codenamen “Spock”)
- als Hauptspeicher in frühen PS/2 Maschinen, z.B. dem ISA-basierten Model 30-286 und dem MCA-basierten 50/60 (nicht den neueren 50Z mit bereits 72-poligen SIMMs).

Diese SIMMs haben eine spezielle Pinbelegung, welche die Erkennung des eingesetzten SIMMs zulässt:

Pin	Standard SIMM	IBM SIMM
2	Data CAS	Data & Parity CAS
24	Not Connected	Presence Detect
26	Parity Data Out	Presence Detect
28	Parity CAS	Not Connected
29	Parity Data In	Parity Data In/Out

Pin 24	Pin 26	Kapazität
offen	offen	kein Modul eingesetzt
offen	GND	512k x 9
GND	offen	256k x 9
GND	GND	1M x 9

Der Speicher (x8) kann mit der Standard-Testfunktion getestet werden.

Das Parity-Bit von 1Mb IBM SIMMs (1M x 1) kann im RCT ab Firmware v25 Beta 5 getestet werden:

3.2.2 Testen von ZIP Speicher

Diese beiden Adapter erlauben es 16 pol. ZIP (64k x 1, 256k x 1) bzw. 20 pol. ZIP (64k x 4, 256k x 4, 1024k x 1, 1024k x 4) RAM zu testen.

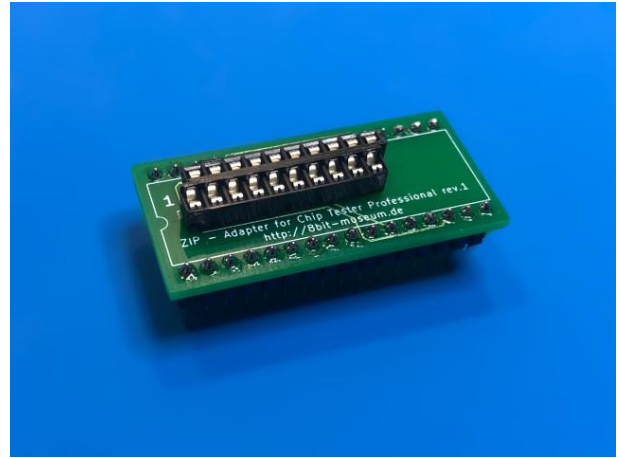


Abbildung 3.13: Adapter für ZIP

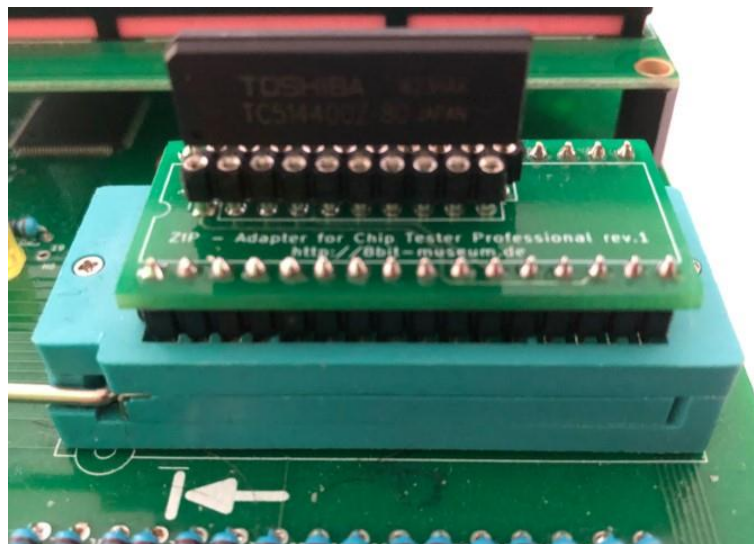


Abbildung 3.14: Adapter im Sockel

Der Chip wird in seinem Adapter in den Tester eingesetzt. Durch Auswahl der korrekten Speichergröße wird der Test („ZIP16 XXX“ oder „ZIP20 XXX“) gestartet.

Der ZIP-Sockel besteht entweder aus zwei 20-poligen Buchenleisten mit gedrehten Kontakten oder aus einem geteilten 20-poligen IC-Sockel mit Federkontakten.

Der Sockel mit Federkontakten ist ggf. die bessere Wahl, da sich hier der Chip leichter wieder entfernen lässt. Der Socket wird hierzu einfach geteilt und die Seitenteile gegeneinander versetzt eingelötet.

Der Sockel ist hierbei wie folgt belegt:

GND	52	1	32	21	Vcc	Din	52	1	32	21	Vcc
A0	51	2	31	22	D0	A0	51	2	31	22	WR
A1	50	3	30	23	D1	A1	50	3	30	23	/RAS
A2	49	4	29	24	D2	A2	49	4	29	24	Dout
A3	48	5	28	25		A3	48	5	28	25	
A4	47	6	27	26		A4	47	6	27	26	
A5	46	7	26	27		A5	46	7	26	27	
A6	45	8	25	28	D3	A6	45	8	25	28	Vss
A7	44	9	24	29		A7	44	9	24	29	
A8	43	10	23	30	/WR	A8	43	10	23	30	TF
A9	42	11	22	31	/RAS		42	11	22	31	nc
	41	12	21	32	/CAS		41	12	21	32	/CAS
	40	13	20	33			40	13	20	33	
/OE	39	14	19	34		A9	39	14	19	34	
	38	15	18	35			38	15	18	35	
	37	16	17	36			37	16	17	36	

Abbildung 3.15: Sockelbelegung bei ZIP20 (256k/1024k x 4) und ZIP20 (1024k x 1)

/OE	52	1	32	21	Vcc	GND	52	1	32	21	Vcc
nc	51	2	31	22	D0	A0	51	2	31	22	Din
A6	50	3	30	23	D1	A1	50	3	30	23	Dout
A5	49	4	29	24	D3	A2	49	4	29	24	
A4	48	5	28	25		A3	48	5	28	25	
A7	47	6	27	26		A4	47	6	27	26	
A3	46	7	26	27		A5	46	7	26	27	
A2	45	8	25	28	Vss	A6	45	8	25	28	
A1	44	9	24	29		A7	44	9	24	29	
A0	43	10	23	30	/WR	A8	43	10	23	30	/WR
	42	11	22	31	/RAS		42	11	22	31	/RAS
	41	12	21	32	/CAS		41	12	21	32	/CAS
	40	13	20	33			40	13	20	33	
D2	39	14	19	34			39	14	19	34	
	38	15	18	35			38	15	18	35	
	37	16	17	36			37	16	17	36	

Abbildung 3.16: Sockelbelegung bei ZIP20 (64k x 4) und ZIP16 (256k x 1)

Als weitere Option ZIP-Speicher zu testen, existieren drei ZIP-to-DIP Adapter.



Abbildung 3.17: ZIP-to-DIP

Den ZIP-2-DIP Adapter gibt es in drei Varianten:

- für 256k x 1 und 64k x 1 ZIP-Module,
- für 64k x 4 und 1024k x 1 ZIP-Module,
- für 256k x 4 und 1024k x 4 ZIP-Module.

Er ermöglicht ebenfalls das Testen von ZIP-Modulen, wie mit dem ZIP20- oder ZIP16-Adapter. Der Unterschied besteht darin, dass die Standard-Testfunktionen für DRAMs verwendet werden.

Aufgrund der Pinbelegung der Speicher im DIP-Gehäuse gibt es hierzu drei Adapter.

Das heißt, es kann z.B. ein 256k x 1 ZIP-Modul mit dem ZIP-to-DIP-Adapter „256k x 1 / 64k x 1“ mit der 41256-Testfunktion getestet werden.

Der Adapter ist nicht dazu geeignet ein ZIP-Modul in einen DIP16/18/20-Sockel einzusetzen, da das Rastermaß 15,24mm beträgt. Dieser Adapter sollte aber auch mit anderen DRAM-Testern funktionieren.

3.2.3 Testen von 7481 bzw. 7484 bipolar SRAM

Es ist möglich das 7481 bzw. 7484 bipolar SRAM (16 Bit x 1) zu testen. Allerdings ist die Ansteuerung nicht ganz trivial, hierfür wird ein spezieller Adapter benötigt.

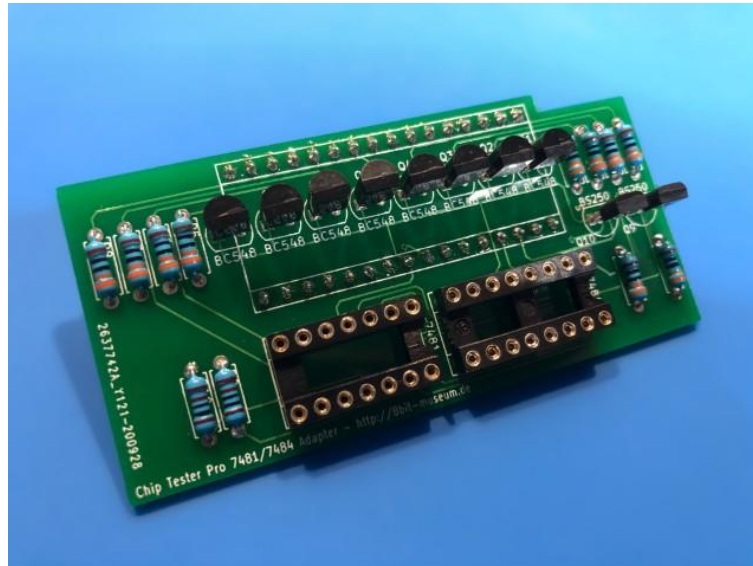


Abbildung 3.18: Adapter für 7481 bzw. 7484



Abbildung 3.19: Adapter im Sockel

3.2.4 Testen von MK4008 und MK4006

Der beiden Speicherchips MK4008 und MK4006 (DRAM, 1024 x 1 Bit) von MOSTEK können mit Hilfe eines einfachen Adapters getestet werden. Der Adapter sorgt dafür, dass an Pin 9 bzw. Pin 10 die Spannungsversorgung von -12V bzw. +5V eingespeist wird. Das 0V Potential wird intern durch einen Spannungsteiler erzeugt.

Die Ausgänge sind nicht TTL kompatibel und benötigen normalerweise einen Treiber. Dennoch kann das IC wie folgt beschrieben getestet werden, wenn auch ein Erfolg nicht immer garantiert werden kann.

Als Adapter kann der spezielle Adapter für den MK4008 verwendet werden oder der Breakout-Adapter.

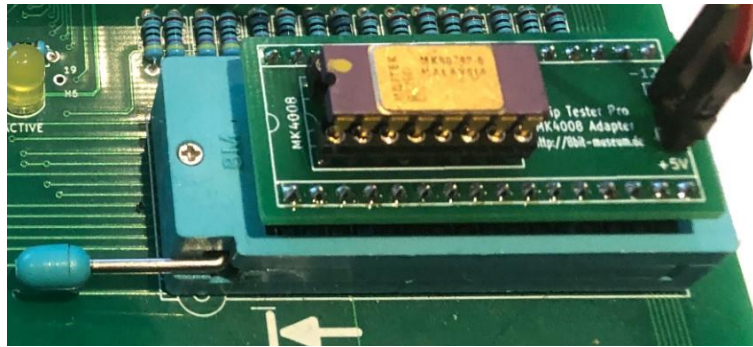


Abbildung 3.20: MK4008 Adapter

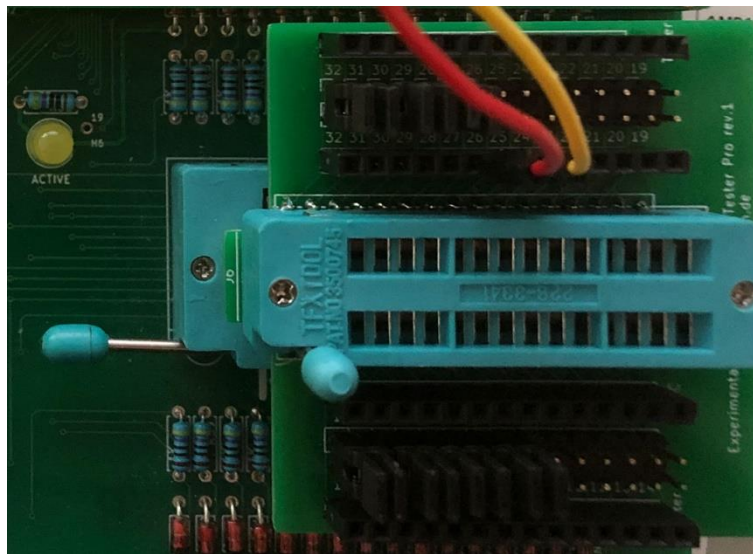


Abbildung 3.21: MK4008 im Breakout-Adapter

Mit Adapter:

Die Spannungsversorgung wird direkt mit der 7pol. Sockelleiste verbunden. Zuerst die +5V, dann die -12V.

Mit Breakout-Adapter:

Die Pins 1-8 und 11-16 (bezogen auf den 16-pol. IC) werden gejumpert. Pin 10 (rotes Kabel) wird mit +5V („5V“ oder “[5V]“ an der 7pol. Sockelleiste) und Pin 11 (gelbes Kabel) mit „-12V“ (auf der DC/DC-Platine verfügbar) verbunden (in dieser Reihenfolge). Es muss ein zusätzlicher 10k Ohm Widerstand an der rechten(!) Seite zwischen dem 5. Pin und 8. Pin gesteckt werden (nicht im Bild, Dout mit 10k Ohm auf GND).

Aufgrund des einfachen Aufbaus kann aber grundsätzlich auch ein einfacher Sockel dazu umgebaut werden.

Zusätzlich wird ein 10k Ohm Widerstand als Pulldown verwendet. Dieser wird an Pin 12 (Dout) des ICs in den ZIF-Sockel gesteckt und mit GND/Vss an der 7pol. Sockelleiste verbunden.

Ab der Firmware v.23 muss der Widerstand wie folgt an den Adapter gelötet werden: An der rechten(!) Seite wird zwischen den 5. Pin und 8. Pin ein 10k Ohm Widerstand gelötet (von oben gezählt, siehe Bild).

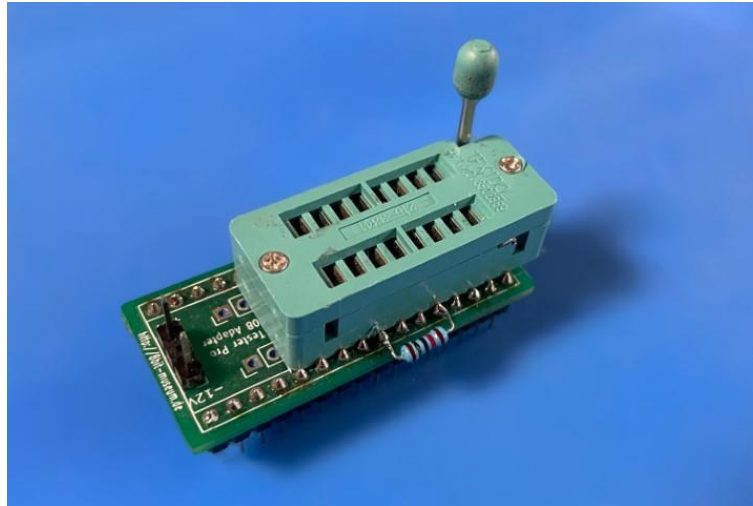


Abbildung 3.22: MK4008 Adapter (modifiziert)

Der aktuelle MK4008 Adapter hat den Widerstand bereits auf der Platine.

3.2.5 Testen von Intel 2107 und kompatible DRAMs

Zum Testen des Intel 2107 und kompatiblen DRAMs wird der „2107 Adapter“ benötigt. Dieser kann ab der Firmware v.20 beta 4 verwendet werden. Der ältere Test bis Firmware v.19 benötigte keinen Adapter, funktionierte erfahrungsgemäß aber nur mit dem Intel 2107C verlässlich.

Der Adapter darf ausschließlich mit dem entsprechenden Menüeintrag „2107 – 2107 Adapter“ verwenden. Mit dem „alten“ Eintrag von Firmware v.19 kann der Adapter zerstört werden.

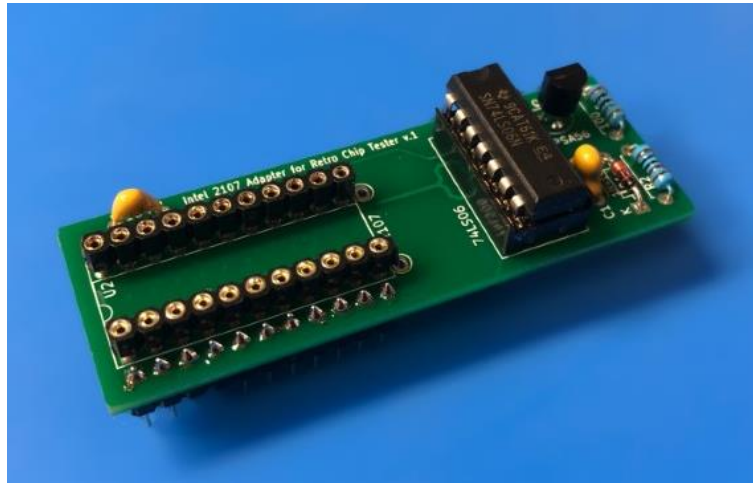


Abbildung 3.23: 2107 Adapter

Beim Testen mit dem Adapter gibt es nicht viel zu beachten. Es wird nur der zu testende IC eingesetzt und der Adapter in den RCT.

Der Adapter wandelt das 5V CE Signal vom RCT in ein 12V CE Signal für den Speicherchip um. Auf keinen Fall der Adapter mit dem alten Test verwendet werden. Es würden 12V an die Eingänge des 7406 gelegt werden und diesen zerstören. Der RCT oder Speicherchip kann dadurch keinen Schaden nehmen.

Technischen Hintergrund:

Idealerweise wird der 2107 mit mehreren Bausteinen angesteuert. Die Erzeugung des CE erfolgt z.B. mit einem Intel 8210 „TTL zu MOS Level Schifter und High Voltage Clock Driver“, der auch gleich die Signalpegel anpasst. Die Ein- und Ausgänge werden über je einen Intel 8212 geführt.

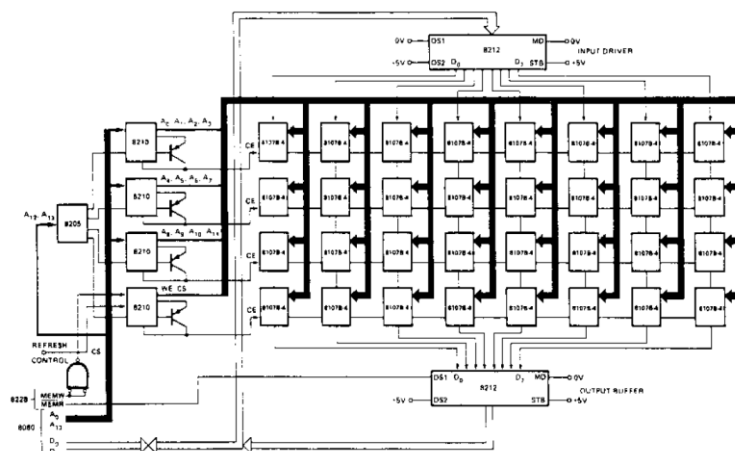


Abbildung 3.24: Ansteuerung Intel 2107

3.2.6 Testen von Intel D8155 und D8156

Es ist möglich den Speicher des Intel D8155 bzw. D8156 (2048-bit Static RAM mit I/O Ports und Timer) mit Hilfe eines Adapters zu testen.

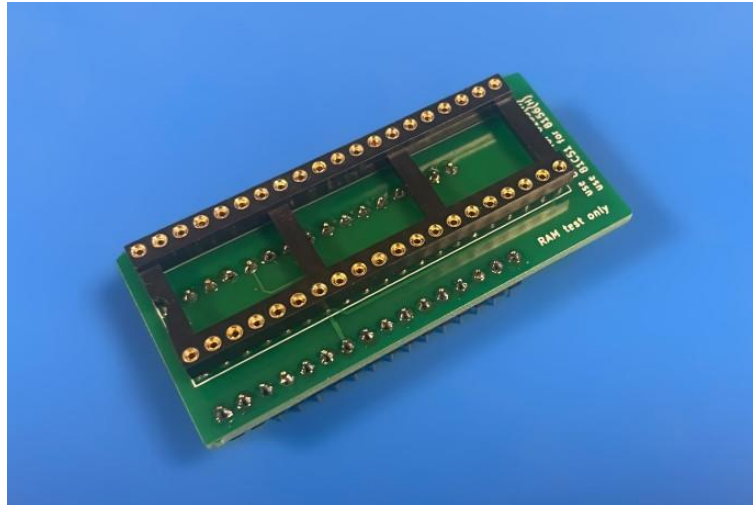


Abbildung 3.25: Adapter für Intel D8155/D8156

Der Test wird mit folgender Einstellung gestartet:

- 81C50 für den Intel D8155 und
- 81C51 für den Intel D8156.

3.2.7 Testen von Motorola MCM6605 und kompatible DRAMs

Zum Testen des MCM6605 und kompatiblen DRAMs wird der „6605 Adapter“ benötigt. Dieser kann ab der Firmware v.24 beta 5 verwendet werden.

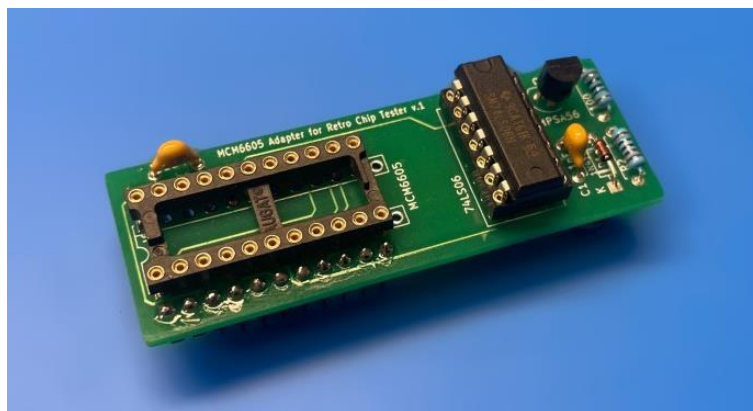


Abbildung 3.26: 6605 Adapter

Beim Testen mit dem Adapter gibt es nicht viel zu beachten. Es wird nur der zu testende IC eingesetzt und der Adapter in den RCT. Der Adapter wandelt das 5V CE Signal vom RCT in ein 12V CE Signal für den Speicherchip um.

3.2.8 Testen von Intel 1101

Der Intel 1101 (SRAM, 256 x 1 Bit) kann mit Hilfe eines einfachen Adapters getestet werden. Der Adapter sorgt dafür, dass an den Pins 4 und 8 die Spannungsversorgung von -9V und an Pin 5 die Versorgungsspannung von +5V eingespeist wird.



Abbildung 3.27: Intel 1101 Adapter

Die Spannungsversorgung wird mit der 7pol. Sockelleiste verbunden. Zuerst GND/Vss, dann die -12V mit dem DC/DC Board.

Zusätzlich muss noch eine „custom definition“ für dieses SRAM installiert werden. Es existiert eine Definition für Dout und das invertierte Dout.

3.2.9 Testen von TMS 4050 und TMS 4051

Es ist möglich die Texas Instruments TMS 4050 und TMS 4051 (4096-bit Dynamic RAM) mit Hilfe eines Adapters zu testen.

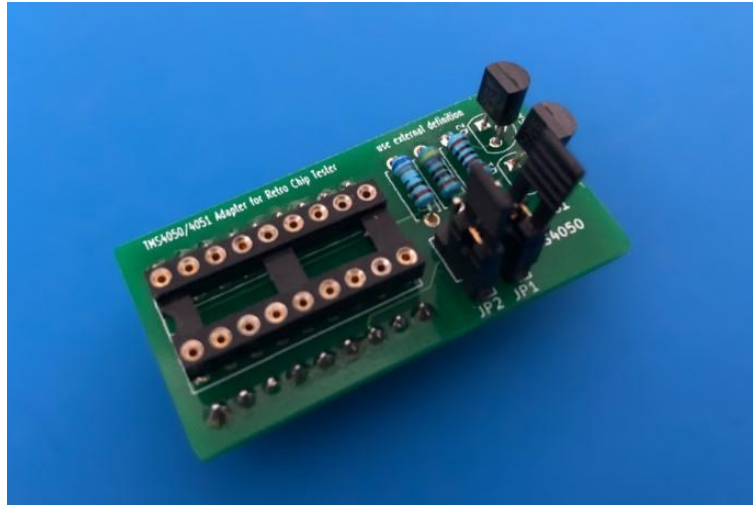


Abbildung 3.28: TMS 4050 / TMS 4051 Adapter

Zusätzlich muss noch eine „custom definition“ für dieses DRAM installiert werden. Es existiert je eine Definition für den TMS 4050 und TMS 4051.

Zum Testen des TMS 4050 müssen beide Jumper nach „unten“ gesetzt werden, zum Testen des TMS 4051 müssen beide Jumper nach „oben“ gesetzt werden.

3.2.10 Selbstdefinierte SRAM, DRAM und ROM Speicherbausteine

Es ist möglich zwölf SRAM, drei DRAM und drei ROM Bausteine selbst zu definieren und für Tests zu verwenden.

Hierzu muss der SRAM-Baustein in der Datei „customsramX.txt“ (X=1, ..., 12 / Slot 1 bis 12) definiert werden und der DRAM-Baustein in der Datei „customdramX.txt“ (X=1, ..., 3 / Slot 1 bis 3). ROM-Bausteine werden in „customromX.txt“ (X=1, ..., 3 / Slot 1 bis 3) definiert. Die Dateien müssen im Hauptverzeichnis der SD-Speicherkarte abgelegt werden.

Nach dem Einschalten des Testers werden die Dateien eingelesen und im EEPROM dauerhaft abgelegt (im Display erscheint kurz „Custom SRAM“, „Custom DRAM“ bzw. „Custom ROM“. Wurde die Datei fehlerfrei eingelesen, wird die Endung zudem von „txt“ in „old“ geändert, damit diese nicht noch einmal eingelesen wird.

Die selbstdefinierten Speicherchips sind über das Menü „CUSTOM“ aufrufbar.

Die selbstdefinierten Speicherchips können über das Konfigurationsmenü gelöscht werden.

Die Konfigurationsdatei für SRAMs (customsramX.txt) muss bzw. kann folgende Einträge beinhalten:

Kommando	Beispiel	Bemerkung
NAME:	2114	Name des Chips (max. 20 Zeichen)
DIP:	18	Anzahl der Pins (14, 16, 18, 20, 22, 24, 28, 32)
SIZE:	1024	Anzahl der Speicherworte
FLAGS:	0	Spezielle Flags (1=DOOUT inverse, 2=OE inverse, 4=WE inverse, 16=WR wird vor CS gesetzt)
A:	5, 6, 7, 4, 3, 2, 1, 17, 16, 15	Adressbus (max. 24 Werte)
DIN:	14, 13, 12, 11	Datenbus Eingabe (max. 16 Werte)
DOOUT:	14, 13, 12, 11	Datenbus Ausgabe (max. 16 Werte)
CS:	8	Chip Select (max. 4 Werte)
PCS:		Positive Chip Select (max. 4 Werte)
WE:	10	Write Enable (1 Wert)
OE:		Output Enable (1 Wert)
GND:	9	0V (max. 4 Werte)
VCC:	18	+5V (max. 4 Werte)
VDD:		+12V (max. 4 Werte)
VBB:		-5V(max. 4 Werte)
CREAD:	AWcPo1DOCP	Read Zyklus (max. 20 Zeichen)
CWRITE:	ADOWcP1CpW	Write Zyklus (max. 20 Zeichen)

Die Konfigurationsdatei für DRAMs (customdramX.txt) muss bzw. kann folgende Einträge beinhalten:

Kommando	Beispiel	Bemerkung
NAME:	4116	Name des Chips (max. 20 Zeichen)
DIP:	16	Anzahl der Pins (14, 16, 18, 20, 22, 24, 28, 32)
SIZE:	16384	Anzahl der Speicherworte
PAGE:	128	Refresh Seitengroße
FLAGS:	0	Spezielle Flags (1=DOUT inverse, 2=OE inverse, 4=WE inverse)
ARAS:	5, 7, 6, 12, 11, 10, 13	Adressbus ROWs (max. 12 Werte)
ACAS:	5, 7, 6, 12, 11, 10, 13	Adressbus COLUMNs (max. 12 Werte)
RAS:	4	Adressbus ROWs select
CAS:	15	Adressbus COLUMNs select
DIN:	2	Datenbus Eingabe (max. 16 Werte)
DOUT:	14	Datenbus Ausgabe (max. 16 Werte)
WE:	3	Write Enable (max. 4 Werte)
OE:		Output Enable (max. 4 Werte)
GND:	16	0V (max. 4 Werte)
VCC:	9	+5V (max. 4 Werte)
VDD:	8	+12V (max. 4 Werte)
VBB:	1	-5V(max. 4 Werte)

Die Konfigurationsdatei für ROMs (customromX.txt) muss bzw. kann folgende Einträge beinhalten:

Kommando	Beispiel	Bemerkung
NAME:	27128	Name des Chips (max. 20 Zeichen)
DIP:	28	Anzahl der Pins (14, 16, 18, 20, 22, 24, 28, 32)
SIZE:	16384	Anzahl der Speicherworte
FLAGS:	0	Spezielle Flags (1=DOUT inverse, 2=OE inverse, 4=WE inverse, 8='1'-padding)
A:	10, 9, 8, 7, 6, 5, 4, 3, 25, 24, 21, 23, 2, 26	Adressbus (max. 20 Werte)
DIN:	11, 12, 13, 15, 16, 17, 18, 19	Datenbus Eingabe (max. 16 Werte), nicht erforderlich
DOUT:	11, 12, 13, 15, 16, 17, 18, 19	Datenbus Ausgabe (max. 16 Werte)
CS:	20	Chip Select (max. 4 Werte)
PCS:		Positive Chip Select (max. 4 Werte)
OE:	22	Output Enable (max. 4 Werte)
GND:	14	0V (max. 4 Werte)
VCC:	1, 27, 28	+5V (max. 4 Werte)
VDD:		+12V (max. 4 Werte)
VBB:		-5V(max. 4 Werte)
CREAD:	AWcPo1DOCp	Read Zyklus (max. 20 Zeichen)

Spezielle Flags („FLAGS:“):

Bit n	Wert	Bit nicht gesetzt	Bit gesetzt	Bemerkung
0	= 1	Dout normal	Dout invertiert	
1	= 2	OE „low active“ (/OE)	OE „high active“ (OE)	
2	= 4	WE „low active“ (/WE)	WE „high active“ (/WE)	
3	= 8	„0“-padding	„1“-padding	nur ROMs kleiner 8 Bit
4	= 16	WE nach CS	WE vor CS	nur SRAM

Der Speicherchip wird generisch angesprochen, d.h. ein SRAM sollte sich wie ein Standard SRAM (z.B. ein 6116) verhalten, ebenso sollte sich ein DRAM wie ein Standard DRAM (z.B. ein 4164) verhalten. Da keinerlei Optimierungen verwendet werden dauert ein solcher Test ca. 3-4x so lange, wie ein vergleichbarer interner Test. Auch können keinerlei Vorgaben an Timings gemacht werden. Es ist also nicht garantiert, dass ein so definierter Chip auch tatsächlich getestet werden.

Wird ein fehlerfreies Testergebnis angezeigt, so ist der Chip vermutlich in Ordnung (zumindest, wenn dieser extrem langsam angesprochen wird).

Schlägt der Test fehl und besitzt man keinen Speicherchip, der damit bereits erfolgreich positiv getestet wurde, kann der Chip defekt sein, er muss es aber nicht und das negative Testergebnis kommt dadurch zustande, dass der Chip sich nicht mit dem generischen Test verträgt.

Beispielkonfiguration eines 2114 SRAM (customsram1.txt):

```
name: 2114 1k x 4 example
dip: 18
size: 1024
flags: 0
a: 5, 6, 7, 4, 3, 2, 1, 17, 16, 15
din: 14, 13, 12, 11
dout: 14, 13, 12, 11
cs: 8
pcs:
we: 10
oe:
gnd: 9
vcc: 18
vdd:
vbb:
```

Beispielkonfiguration eines 4116 DRAM (customdram1.txt):

```
name: 4116 16k x 1 example
dip: 16
size: 16384
page: 128
flags: 0

aras: 5, 7, 6, 12, 11, 10, 13
acas: 5, 7, 6, 12, 11, 10, 13
din: 2
dout: 14
ras: 4
cas: 15
we: 3
oe:

gnd: 16
vcc: 9
vdd: 8
vbb: 1
```

Falls die generischen Schreib-/Lesezyklen mit einem IC nicht kompatibel sind, z.B. wenn die Adresse und Daten getaktet abgelegt werden müssen, können diese mit den Befehlen CREAD und CWRITE selbst definiert werden. Diese Definition erlaubt kein genaues Timing und ist gegenüber der „normalen“ Tests sehr langsam. Aus diesem Grund ist diese Definition auch nur bei SRAMs und ROMs möglich. Von den Flags werden nur „DOUT inverse“ und „1'-padding“ beachtet.

Zeichen	Bedeutung (CREAD und CWRITE)
A	Anlegen der unteren Adresse (Adressbreite wie in "a:" definiert)
a	Anlegen der oberen Adresse (Adressbreite wie in "a:" definiert) ³
D	Schreiben der Daten (CWRITE), Lesen der Daten (CREAD)
C	alle CS auf HIGH
c	alle CS auf LOW
P	alle PCS auf HIGH
p	alle PCS auf LOW
W	WE auf HIGH
w	WE auf LOW
O	OE auf HIGH
o	OE auf LOW
1..4	Verzögerung von 60/120/180/240ns

³ z.B. verwendet bei RCA CDP1831 und CDP1833

3.3 Problematische Speicherbausteine

In diesem Kapitel werden einige „problematische“ Speicherbausteine beschrieben und Möglichkeiten vorgestellt, wie diese geprüft werden können. Es gibt aber keine Garantie, dass diese „Workarounds“ immer funktionieren. Aufgrund der Flexibilität die der Tester besitzt, muss man diese (geringen) Einschränkungen leider in Kauf nehmen.

3.3.1 Testen von älteren DRAMs

I.d.R. können alle DRAMs problemlos getestet werden. Leider gibt es einige wenige DRAMs, hauptsächlich der Typen 2104er (4k x 1), 4116 (16k x 1), 3732 (32k x 1), 4164 (64k x 1), 4416 (16k x 4) und teilweise 4464 (64k x 4), bei denen ein Test teilweise direkt am Anfang (an Adresse „0“) und kurz darauf mit einem Fehler abbricht (oder die erst gar nicht erkannt werden). Hier gibt es mehrere Möglichkeiten, warum ein Test nicht durchgeführt werden kann, aber auch entsprechend viele Lösungen:

1. Einige Typen haben eine erhöhte Stromaufnahme die mit einem Blockkondensator abgefangen werden muss (meistens >500mW). Hierzu einen 100 nF Kondensator zwischen Vcc und Vss mit in den Sockel klemmen. Bei einem 4116 sind dieses Pin 9 (Vcc) und Pin 16 (Vss). Bei einem 4164 sind dieses Pin 8 (Vcc) und Pin 16 (Vss).

Hierfür kann auch der Entkopplungsadapter (siehe Kapitel 3.3.4) verwendet werden (den entsprechenden Jumper setzen) oder der „on-board“ Entkopplungskondensator.

2. Durch die erhöhte Stromaufnahme fällt an den Transistoren eine höhere Spannung ab, so dass gerade im Einschaltmoment die Versorgungsspannung stark abfällt (ein einfacher Blockkondensator reicht hier nicht mehr). In diesem Fall Vcc des ICs mit Vcc (5V) an der oberen Sockelleiste verbinden.
3. Einige ältere DRAMs benötigen Pullups an den Datenleitungen, damit die Signalpegel ausreichend sind. Hierzu in der Konfiguration die standardmäßig ausgeschalteten Pullups probeweise einschalten (Config → „DRAM Pullup: 1“.)

Werden DRAMs der o.g. Typen getestet, kann wie folgt vorgegangen werden:

Zunächst die DRAMs ohne Entkopplungskondensator testen. Werden diese als OK getestet, dann sind diese bzgl. ihres Verhaltens als „sehr gut“ zu klassifizieren. Die ICs, die ohne Entkopplungskondensator als defekt getestet wurden, werden anschließend noch einmal mit Entkopplungskondensator getestet. Die ICs, die so als OK getestet wurden, sind im Rahmen der Spezifikation ebenfalls i.O., besitzen aber höhere Einschaltströme oder erzeugen aufgrund von Stromspitzen Störungen auf der Vcc Leitung. Alle ICs, die auch mit Entkopplungskondensator als defekt getestet wurden, sollten aussortiert werden.

3.3.2 Testen von ICs mit einer erhöhten Stromaufnahme

Der Tester ist in der Lage ICs mit einer Peak-Leistungsaufnahme von 1 Watt (und mehr) zu testen. Bei einigen wenigen ICs (vor allem bei ICs mit Alterungserscheinungen) tritt aber gerade zu Anfang ein erhöhter Strombedarf auf, der zu Störungen führen kann.

Sollte bei einem älteren IC direkt am Anfang (Adresse 0) der Test mit einem Fehler beendet werden oder der IC im Sockel erst gar nicht erkannt werden, dann können folgende Maßnahmen ggf. helfen:

- Hinzufügen eines 100nF Kondensators (Blockkondensator) zwischen GND und Vcc (zusammen mit dem IC in den Sockel klemmen). Dieses überbrückt die plötzlich abfallende Spannung.
- Verbinden von Vcc und/oder GND direkt mit der Spannungsversorgung des Testers (Sockel zwischen Display und DC/DC-Modul). Dieses verhindert einen erhöhten Spannungsabfall an den Transistoren aufgrund eines erhöhten Strombedarfs.

I.d.R. sollte der zusätzliche Kondensator ausreichen. In alten Schaltungen sind diese Blockkondensatoren an jedem IC zu finden. Leider kann aufgrund der unterschiedlichen Pinbelegung der zu testenden ICs ein solcher Kondensator nicht grundsätzlich eingebaut werden.

Sollte auch mit diesen Maßnahmen der Test nicht durchlaufen, sollte der IC nicht mehr verwendet werden.

3.3.3 Testen der Ausgangstreiber von ICs

Bei älteren ICs kann es vorkommen, dass die Ausgangstreiber nicht mehr in der Lage sind den im Datenblatt angegebenen Strom zu liefern (oft zwischen 5 - 20mA). In einem realen System treten dann Fehler auf, da der Ausgang nicht mehr den benötigten Strom liefern kann. Im Tester kann dieser Baustein als fehlerfrei getestet werden, da die Ausgänge durch die (MOS) Eingänge der MPU quasi nicht belastet werden (nur ein paar uA → spannungsgesteuert).

Wer die Ausgänge unter Last testen will, muss einen externen Widerstand anlegen. Dieses kann bequem durch den Breakout-Adapter geschehen oder durch einklemmen eines entsprechenden Widerstands im ZIF-Sockel.



Abbildung 3.29: Lastwiderstand

Der im Bild gezeigte NEC uPD4164 sollte 5mA liefern können. Ein 1k Ohm Widerstand belastet den Ausgang mit ca. 5mA. Der Test sollte bei funktionsfähigen Ausgangstreibern weiterhin fehlerfrei durchlaufen.

Soll der Ausgang mit 10mA belastet werden, kann ein 470 Ohm Widerstand verwendet werden. Bitte das Datenblatt konsultieren, wie hoch der Ausgangsstrom maximal sein darf, damit der Baustein nicht beschädigt wird.

3.3.4 Verwenden des Entkopplungsadapters (Decoupling Adapter)

Einige DRAMs haben einen erhöhte Stromaufnahme die mit einem Blockkondensator abgefangen werden muss. Hierzu kann ein 100 nF Kondensator zwischen Vcc und Vss mit in den Sockel geklemmt werden. Anstatt einen 100 nF Kondensator zwischen Vcc und Vss in den Sockel einzuklemmen, kann der Entkopplungsadapter verwendet werden.

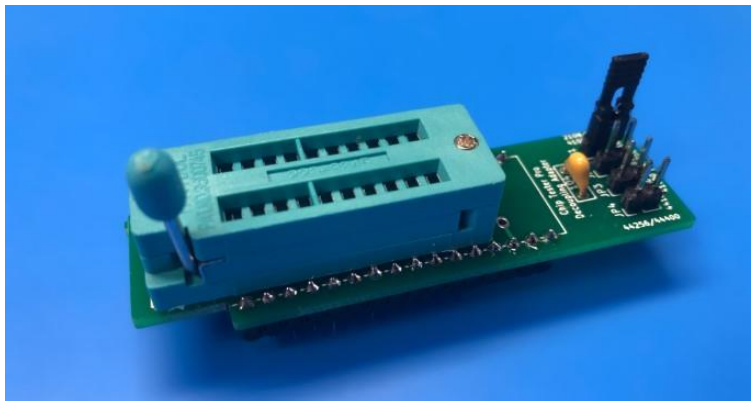


Abbildung 3.30: Entkopplungsadapters

Normalerweise wird dieser Adapter nicht benötigt, da Vcc zuverlässig zur Verfügung steht. Wenn ein DRAM einen erhöhten Stromverbrauch hat (meist > 500mW) und ein Test sofort fehlschlägt, kann dieser Adapter verwendet werden.

Dieser Adapter kann u.a. mit folgenden ICs verwendet werden:

Gehäuse	Vcc Pin ⁴	Jumper	Unterstützte ICs (und andere mit gleicher Pinbelegung)
DIP16	9	JP1	DRAMs, wie 4116, 4108, 2108, 2104
DIP16	8	JP2	DRAMs, wie 4816, 4164, 3732, 4128, 41256 SRAMs, wie 3101, 2112 SRAMs, wie 6561 (DIP18(!)) und die meisten Logik-ICs
DIP18	9	JP3	DRAMs, wie 4408, 4416, 4464, 41000 SRAMs, wie 74c910, 6518, 2114 und die meisten Logik-ICs
DIP20	10	JP4	DRAMs, wie 44256, 44400 SRAMs, wie 4047 und die meisten Logik-ICs

Abbildung 3.31: Unterstützte ICs

Der Entkopplungskondensator kann natürlich auch bei vielen SRAMs und Logik-ICs im DIP16, DIP18 und DIP20 Gehäuse verwendet werden, auch wenn es hier nicht explizit notwendig ist.

Das Board Rev.1.2k hat die gleichen Jumper neben dem ZIF-Sockel, so dass diese anstelle des Entkopplungsadapters verwendet werden können.

⁴ Vss bei SRAMs und Logik-ICs

Der Entkopplungskondensator kann über vier Jumper eingestellt werden:

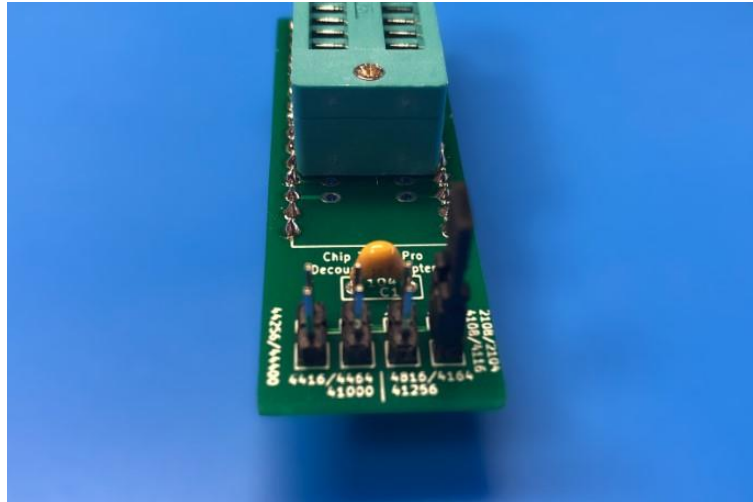


Abbildung 3.32: Jumper

Die 1x03-Stiftleiste wird verwendet, um den IC direkt mit Strom zu versorgen. Wenn ein IC viel Strom benötigt, kann er direkt mit dem Socket (zwischen DC/DC-Modul und Display) verbunden werden. Der mittlere Stift kann abgeschnitten oder entfernt werden (er ist nicht angeschlossen).

J4 kann dafür verwendet werden, um Vcc (links) und Vss (rechts) zu injizieren, wenn ein IC hohen Strom benötigt. Dies scheint bei den bereits getesteten ICs nicht notwendig zu sein, aber die Platine ist zumindest dafür vorbereitet (die Transistoren können so umgangen werden und die Power-Pins des ICs direkt mit der Strombuchse zwischen dem DC/DC-Modul und der Anzeige verbunden werden).

3.3.5 Verwenden des „on-board“ Entkopplungskondensators (ab Rev.1.2k)

Direkt neben dem ZIF-Sockel befinden sich drei Stiftleisten mit denen ein 100nF Entkopplungskondensator bei DRAM-Tests zugeschaltet werden kann.

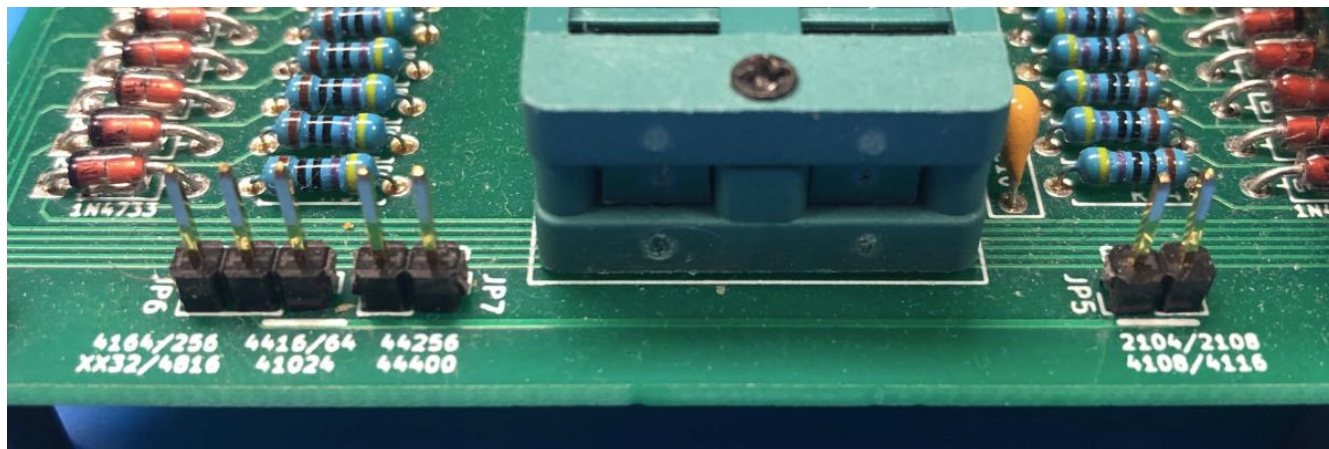


Abbildung 3.33: „on-board“ Entkopplungskondensator

Die unterstützten ICs sind dieselben wie beim Entkopplungsadapter:

Gehäuse	Vcc Pin ⁵	Jumper	Unterstützte ICs (und andere mit gleicher Pinbelegung)
DIP16	9	JP5	DRAMs, wie 4116, 4108, 2108, 2104
DIP16	8	JP6 links	DRAMs, wie 4816, 4164, 3732, 4128, 41256 SRAMs, wie 74c910, 3101, 2112 SRAMs, wie 6561 (DIP18(!)) und die meisten Logik-ICs
DIP18	9	JP6 rechts	DRAMs, wie 4408, 4416, 4464, 41000 SRAMs, wie 6518, 2114 und die meisten Logik-ICs
DIP20	10	JP7	DRAMs, wie 44256, 44400 SRAMs, wie 4047, 2114 und die meisten Logik-ICs

Abbildung 3.34: Unterstützte ICs

Der Entkopplungskondensator kann natürlich auch bei vielen SRAMs und Logik-ICs im DIP16, DIP18 und DIP20 Gehäuse verwendet werden, auch wenn es hier nicht explizit notwendig ist.

⁵ Vss bei SRAMs und Logik-ICs

3.3.6 Testen von ITT4027 (und ggf. kompatible SRAMs)

Der Speicherbaustein ITT4027 (und vermutlich auch alle kompatiblen und von der Stromaufnahme her vergleichbaren SRAMs) können Probleme mit der geschalteten Spannung Vss haben.

Um diese Bausteine zu testen, muss Pin 16 (Vss) direkt mit Vss/GND verbunden werden. Hierzu Pin 16 mit Vss an der 7pol. Pfostenleiste verbinden.

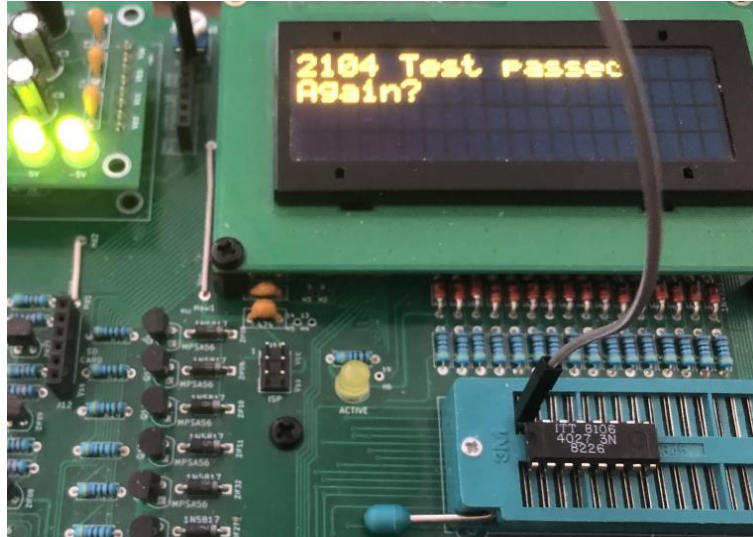


Abbildung 3.35: Testen des ITT4027

Alternativ kann ein (Folien-)Kondensator von 100nF zwischen Vss (Pin 16) und Vcc (Pin 9) geklemmt werden (Entkopplungskondensator) aufgrund der hohen Stromaufnahme.

3.3.7 Testen von SIMM/SIPP Modulen

Der Tester kann SIMM/SIPP Module testen. Hier ist aber die maximale Stromaufnahme der Module zu beachten. Der Tester liefert maximal bis zu ca. 300-400mA. Sofern die maximale Stromaufnahme der Module ungefähr in diesen Grenzen bleibt, sollten diese testbar sein.

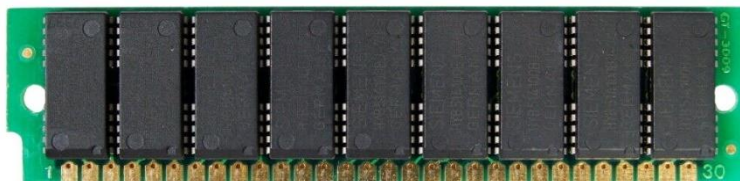


Abbildung 3.36: SIMM 30 (9 ICs)

Die folgenden Datenblätter zeigen in etwa wo die Grenzen liegen:

Operating Current* (RAS, CAS, Address Cycling @ t _{RC} =min.)	KMM594000A- 7	I _{CC1}	—	945	mA
	KMM594000A- 8		—	855	mA
	KMM594000A-10		—	765	mA

Das Modul (mit neun ICs) ist nicht mehr verlässlich testbar (bei den Werten handelt es sich um maximale Werte, die tatsächliche Stromaufnahme liegt niedriger). Es gilt: Je schneller das Modul und je mehr ICs, desto höher die Stromaufnahme.

Das folgende Modul (mit drei ICs) ist testbar:

Operating Current* (RAS, CAS, Address cycling @t _{RC} =min.)	KMM594100AN - 5	I _{CC1}	-	305	mA
	KMM594100AN - 6		-	275	mA
	KMM594100AN - 7		-	245	mA
	KMM594100AN - 8		-	215	mA

3.3.8 DRAMs aus russischer bzw. osteuropäischer Fertigung

Sollten Speichertests von DRAMs aus russischer bzw. osteuropäischer Fertigung häufig fehlschlagen, könnte es an zu geringen Spannungspegeln für ein logisch High aufgrund des Alters der Bausteine liegen. Hierzu können die Pullup-Widerstände für DRAMs in der Konfiguration probeweise eingeschaltet werden. Damit wird der High-Pegel etwas angehoben, so dass der Test ggf. positiv durchläuft.

Jedoch sollte hierbei beachtet werden, dass, wenn diese Chips noch in einem Computer verwendet werden, der Pegel vermutlich auch dort zu niedrig oder zumindest „hart an der Grenze“ ist. Eine verlässliche Funktion über einen längeren Zeitraum ist hier vermutlich nicht mehr gegeben.

3.4 Testen von ROM, PROM, EPROM oder EEPROM

Der Retro Chip Tester kann eine Vielzahl von ROMs, PROMs, EPROMs und EEPROMs testen bzw. auslesen. Das Testen beruht darauf, dass über den Inhalt eine CRC32 gebildet wird, die man mit einer bereits bekannten Prüfsumme vergleichen kann. Ist diese in der internen Datenbank vorhanden so wird der Inhalt direkt angezeigt. Ist eine SD-Speicherkarte angeschlossen, kann der Inhalt auch ausgelesen und gespeichert werden.

Bei der Auswahl des Speicherchips werden einige Zusatzinformationen im Display angezeigt:



Abbildung 3.37: EPROM 2708

In der vierten Zeile werden i.d.R. Infos zur Spannungsversorgung angezeigt. Beim EPROM 2708 sind dieses Vss, Vcc, Vbb und Vdd. Da die Größe des Displays begrenzt ist, werden hier nur Vdd (12V an Pin 19) und Vbb (-5V an Pin 21) angezeigt.

Bei einigen EPROMs, z.B. dem 2764, werden **während des Tests** zusätzlich besondere Beschaltungen angezeigt, z.B. Vcc an Pin 1 (Vpp) und Vcc an Pin 27 (/PGM).

Im Fortschrittsbalken werden leere „Blöcke“ (1 Block = 1/16 des Gesamtspeichers eines Chips) mit einem leeren Block angezeigt. So kann man schnell erkennen, ob ein Speicher z.B. halb leer ist.

Eine Speicherzelle wird als „leer“ angesehen, wenn der Inhalt 0xFF (oder 0xF) ist. Das gilt auch für bipolare PROMs, die standardmäßig 0x00 beinhalten.

3.4.1 Identifizieren von Inhalten

Wird ein ROM, PROM, EPROM oder EEPROM eingelegt, kann eine CRC32 berechnet werden. Damit lässt sich feststellen, ob der Inhalt grundsätzlich noch korrekt gelesen werden kann. Seit der Firmware v19 wird die Datenbank auf der SD Karte abgelegt.

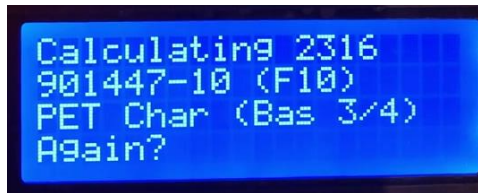


Abbildung 3.38: 2316 ROM erkannt als „901447-10“ für den Commodore PET

Sollte bei einem ROM die CRC32 bekannt sein, wird dessen Bezeichnung ausgegeben, ansonsten die berechnete CRC32. Wird eine CRC32 ausgegeben, sollte der Test mehrmals wiederholt werden. Ändert sich dabei die CRC32 mehrmals, ist der Chip mit hoher Wahrscheinlichkeit defekt oder die Versorgungsspannung zu gering. In diesem Fall den Tester probeweise per Hohlstecker mit Spannung versorgen.

Ab Firmware v.27 ist diese Funktion verbessert:

Enthält die Datenbank mehrere Einträge zu einer CRC32, weil der Baustein in verschiedenen Systemen verwendet wurde, zeigt der RCT nun alle Treffer an. Für die CRC32 0xe47045f4 werden somit beispielsweise folgende Einträge angezeigt werden:

342-0132-c.e12;Apple //c
342-0132-c.e12;Apple //c (Original-Speichererweiterung)
342-0132-c.e12;Apple //c (UniDisk 3.5)
342-0132-c.e12;Apple //c (Rev. 4)
342-0132-c.e12;Apple //e
342-0132-c.e12;Franklin ACE 2200
342-0132-c.e12;Franklin ACE 500
342-0132-c.e12;Laser 128
342-0132-c.e12;Laser 128 (Original-Hardware)
342-0132-c.e12;Laser 128ex (Version 4.5)
342-0132-c.e12;Laser 128ex2 (Version 6.1)
342-0132-c.e12;Mikroprofessor III
342-0132-c.e12;Pravetz 8C
342-0132-c.e12;Spectrum ED
342-0132-c.e12;TK3000//e

Drücken Sie die „beliebige“ Taste, um den nächsten Eintrag anzuzeigen, und „TOP“, um den Vorgang zu beenden. Alle Einträge werden auch in der zusätzlichen Textdatei auf der SD-Karte gespeichert, wenn der Inhalt gesichert wird.



Die Berechnung der CRC32 über große Speicherbausteine, wie z.B. 271000 / 231000 und darüber (ab 128k x 8), ist extrem langsam. Die Möglichkeit wurde eingebaut, um in seltenen Fällen eine Checksumme ermitteln zu können, nicht um Serientests damit durchzuführen.

3.4.2 Auslesen von ROM, PROM, EPROM oder EEPROMs

Mit einem geeigneten Micro-SD Kartenadapter können Speicherbausteine auch ausgelesen werden.

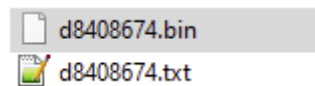
Hierzu muss in der Konfiguration die Unterstützung für SD-Karten eingeschaltet werden ("**SD Card: 1**") und der Micro-SD Kartenadapter muss mit dem Chip-Tester verbunden werden. Die Konfiguration ist in Kap. 5 beschrieben.

Wird bei eingeschalteter Unterstützung...

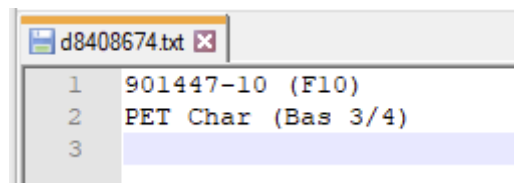
... **die OK Taste lang gedrückt**, so wird der Inhalt des Speicherbausteins in die Datei `<crc32>.bin` geschrieben. Ist das ROM bekannt, wird eine weitere Datei `<crc32>.txt` erstellt, die eine Beschreibung zum Speicherbaustein enthält.

Beispiel: PET Character ROM für Basic 3 und 4

Es werden zwei Dateien erzeugt:



Der Inhalt der Text-Datei lautet:



... **die OK Taste kurz gedrückt**, wird die nur die CRC32 berechnet und ggf. der erkannte Speicherbaustein angezeigt.

Besitzt ein Speicher weniger als 8-Bit Datenbusbreite, wird standardmäßig mit ,0' gepaddet. In der Konfiguration kann das Padding auf ,1' geschaltet werden. Das Padding wird in die CRC32 Berechnung mit einbezogen.



Die Speicherkarte muss FAT16 oder FAT32 formatiert sein. Falls sie nicht erkannt wird, ggf. einmal mit dem SD Memory Card Formatter der SD Association formatieren.

Download unter: <https://www.sdcard.org/downloads/formatter/>



Da der Tester über keine Echtzeituhr verfügt, wird standardmäßig kein Dateidatum gesetzt. Es gibt jedoch die Möglichkeit ein Datum mit Uhrzeit vorzugeben, indem man auf der SD-Karte eine Datei `datetime.txt` anlegt. Wird der Tester eingeschaltet (bzw. nach einem Reset), werden Datum und Uhrzeit dieser Datei beim Anlegen von weiteren Dateien verwendet.

3.4.3 Vergrößern der internen ROM-Datenbank

Der Chip Tester besitzt eine interne Datenbank mit einigen ROMs. Diese Liste kann über eine externe Datei auf der SD-Speicherkarte erweitert werden.

Die Datei muss den Namen „customcrc32.txt“ tragen und wie folgt aufgebaut sein:

```
; Das ist ein Kommentar  
<crc32>; <rom name>; <rom description>
```

Als Beispiel:

```
0xd8408674;901447-10 (F10);PET Char (Bas 3/4)
```

Die CRC32 muss den Prefix „0x“ besitzen. Es werden nur jeweils die ersten 40 Zeichen beachtet, von denen nur die ersten 20 Zeichen auf dem Display angezeigt werden können.

Damit die Datenbank schnell durchsucht werden kann, muss mit dem beiliegenden Compiler (customcrc32.exe) eine Indexdatei (customidx32.bin) erstellt werden, die zusammen mit der Datenbank auf der SD-Speicherkarte abgelegt wird.

3.4.4 Verwenden von vorbereiteten Datenbanken

Es gibt bereits einige vorbereitete Datenbanken zum Download:

-  [C64 Carts \(2020-12-26\)](#)
-  [Full List \(2022-02-25\)](#)
-  [Game Carts \(2021-08-05\)](#)
-  [Index Compiler \(Win64\)](#)
-  [MAME 0.233 full list \(2021-07-23\)](#)
-  [MAME 0.233 no duplicates \(2021-07-23\)](#)
-  [NES Carts \(2017-08-21\)](#)
-  [PinMAME 3.3b \(2021-01-01\)](#)
-  [Synthesizer ROMs CRC32 \(2020-11-06\)](#)

Die umfangreichste Datenbank ist die „Full List“ mit ca. 400.000 Einträgen, die alle anderen Datenbanken beinhaltet. Es spricht im Grunde nichts dagegen grundsätzlich diese Datenbank zu verwenden, man sollte aber einen Nachteil des CRC32-Verfahrens nicht vergessen: Es können Kollisionen auftreten, d.h. ROMs mit unterschiedlichem Inhalt können durchaus dieselbe CRC32 Prüfsumme aufweisen. Die Wahrscheinlichkeit liegt bei etwa 1:10.000. Wer nur mit Synthesizer ROMs arbeitet, sollte lieber die spezialisierte Datenbank verwenden, die mit nur 2000 Einträgen die Wahrscheinlichkeit einer Kollision auf etwa 1:2.000.000 reduziert.

Hat man sich für eine Datenbank entschieden, werden die beiden Dateien

customcrc32.txt und customidx32.bin

einfach in das Hauptverzeichnis der SD-Karte kopiert. Beim nächsten Lesen eines ROMs (kurzer Druck auf OK) oder Speichern eines ROMs (langer Druck auf OK), wird der Inhalt anhand dieser Datenbank identifiziert.

Hinweis:

Einige ROMs werden von mehreren Systemen verwendet, z.B. wird folgendes Keyboard-ROM in 15 Systemen verwendet (aus „MAME x.xxx full list“):



- 0xe47045f4;342-0132-c.e12;Apple //c
- 0xe47045f4;342-0132-c.e12;Apple //c (Original Memory Expansion)
- 0xe47045f4;342-0132-c.e12;Apple //c (UniDisk 3.5)
- 0xe47045f4;342-0132-c.e12;Apple //c (rev 4)
- 0xe47045f4;342-0132-c.e12;Apple //e
- 0xe47045f4;342-0132-c.e12;Franklin ACE 2200
- 0xe47045f4;342-0132-c.e12;Franklin ACE 500
- 0xe47045f4;342-0132-c.e12;Laser 128
- 0xe47045f4;342-0132-c.e12;Laser 128 (original hardware)
- 0xe47045f4;342-0132-c.e12;Laser 128ex (version 4.5)
- 0xe47045f4;342-0132-c.e12;Laser 128ex2 (version 6.1)
- 0xe47045f4;342-0132-c.e12;Microprofessor III
- 0xe47045f4;342-0132-c.e12;Pravetz 8C
- 0xe47045f4;342-0132-c.e12;Spectrum ED
- 0xe47045f4;342-0132-c.e12;TK3000//e

Der RCT kann nur einen Eintrag anzeigen und zeigt den ersten gefundenen Eintrag an (in der größenoptimierten Liste „Full list“ wurde doppelte Einträge bereits entfernt).

Ein 342-0132-c.e12 ROM aus einem „Laser 128“ kann somit als „Apple IIc“ angezeigt werden (der Inhalt ist identisch, da gleiche CRC32). Wer sich unsicher ist, sollte die CRC32 noch einmal mit Hilfe eines Texteditors in der „MAME x.xxx full list“ suchen und nachsehen, welche Systeme dieses ROM tatsächlich verwenden.

Ein weiterer Hinweis:

Es wird eine CRC32 verwendet, die nicht vergleichbar mit einem Hash ist, wie z.B. SHA1 oder SHA256. Die Wahrscheinlichkeit einer Kollision ist zwar gering, bei 400000 Einträgen ist es aber durchaus sehr wahrscheinlich, dass eine CRC32 verschiedene Inhalte beschreibt. Für Mathematiker: Ab 77163 Einträgen liegt die Wahrscheinlichkeit einer Kollision bei 50%.

3.4.5 Besonderheiten beim Auslesen von ROMs und PROMs

Beim Auslesen von ROMs ist eine Besonderheit zu beachten: Bei den Signalen „Chip Select“, „Chip Enable“ und „Output Enable“ Signale kann bei der Herstellung festgelegt werden, ob diese Low-Aktiv oder High-Aktiv sein sollen. Dieses hat natürlich Auswirkungen, wenn der auszulesende Chip den erwarteten Signalpegel nicht verwendet. Der Chip Tester unterstützt die häufigsten vorkommenden Signalpegelkombinationen:

ROM	Typ	Pin - Signal	aktiv	zum Beispiel verwendet in
2308 - 1k x 8	A	18 – /CS2* 20 – /CS1	low low	
	B	18 – /CS2* 20 – CS1	low high	
	C	Autodetect	high/low	
2316 - 2k x 8	A	18 – /CS2* 20 – /CS1* 21 – CS3*	low low high	Commodore
	B	18 – CS2* 20 – /CS1* 21 – /CS3*	high low low	Apple II
	C	Autodetect	high/low	
2332 - 4k x 8	A	20 – /CS1* 21 – CS2*	low high	Acorn Atom Commodore CBM, C64
	B	20 – /CS1* 21 – /CS2*	low low	TI 99/4A Sinclair ZX80
	C	20 – CS1* 21 – CS2*	high high	Gottlieb Pinball Machines
	D	Autodetect	high/low	
2364(24) - 8k x 8	A	20 – /CS*	low	Commodore 1541, VC20 Atari 400/800, DAI EACA Video Genie Sinclair ZX81, Vtech VZ 200
	B	20 – CS*	high	
	C	Autodetect	high/low	
2364(28) - 8k x 8	A	20 – /CE 22 – /OE 26 – CS2 27 – CS1	low low high high	VEB Robotron KC87/KC85 Apple Lisa, CoCo 2 Vtech VZ 300 (Memotech MTX)
	B	Autodetect	high/low	
23128 - 16k x 8	A	20 – /CS2* 22 – /CS1* 27 – CS3*	low low high	Commodore 16/116, Plus/4 Commodore C64/C128 BBC B, Matra Alice 32, Oric I
	B	20 – /CS2* 22 – /CS1* 27 – /CS3*	low low low	Sinclair ZX Spectrum BBC B, Dragon 32/64
	C	20 – CS2* 22 – OE* 27 – /CS3*	high high low	Sinclair QL
	D	Autodetect	high/low	

ROM	Typ	Pin - Signal	aktiv	zum Beispiel verwendet in
23256 - 32k x 8	A	20 – /CE* 22 – /OE*	low low	Amstrad CPC, Atari ST Commodore C128 Enterprise 64, CoCo 3
	B	20 – /CE* 22 – OE*	low high	Sinclair QL
	C	Autodetect	high/low	
23512 - 64k x 8		20 – /CE 22 – /OE	low low	
231000 - 128k x 8		20 – /OE*	low	
232000 - 256k x 8		22 – /CE 24 – /OE	low low	
234000 - 512k x 8		22 – /CE 24 – /OE	low low	
6540 - 2k x 8		2 – /CS5 3 – /CS4 4 – /CS3 17 – CS1 27 – CS2	low low low high high	Commodore PET

*) Signal (High- oder Low-Aktiv) programmierbar

Werden bipolare (P)ROMs ausgelesen, ist zu beachten, dass der Chip Tester die ausgelesenen Daten nicht bearbeitet und so abgelegt, wie sie ausgelesen wurden. Das ist insofern wichtig, wenn diese dazu verwendet werden sollen ein neues PROM zu programmieren, da die Daten bei einigen PROMs hierfür invertiert vorliegen müssen. I.d.R. sollte die Software des Programmierers dieses regeln.

Da einige (P)ROMs über Open Collector Ausgänge verfügen, sollte sichergestellt sein, dass für diese Typen die Pull-ups aktiviert (Standard) sind.

Bei 4-bittigen Speicher, werden die vier Bit in den unteren vier Bits eines Bytes abgelegt. Wird also ein PROM mit 256x4 Bit ausgelesen, wird auf der SD-Speicherkarte eine Datei mit 256 Byte abgelegt, die in den jeweils unteren 4-Bit eines jeden Bytes die Daten enthält.

Wird ein ROM ausgewählt, bei dem mehrere Profile hinterlegt sind, so kann dieses anschließend ausgewählt werden. Das Display zeigt dann hinter „L=“ die Pins an, die low aktiv sind, und hinter „H=“ die Pins, die high aktiv sind. In der dritten Zeile wird eine kleine Orientierung ausgegeben

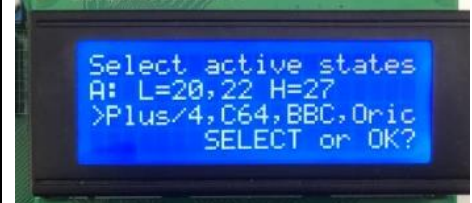
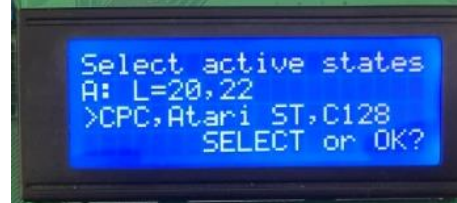
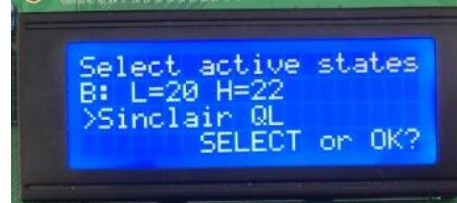
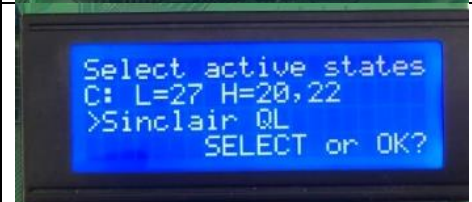
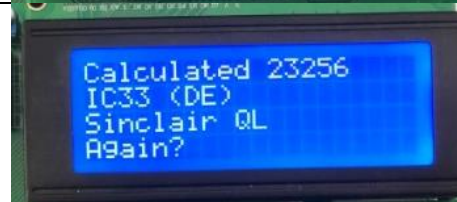
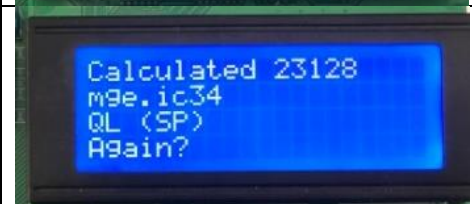
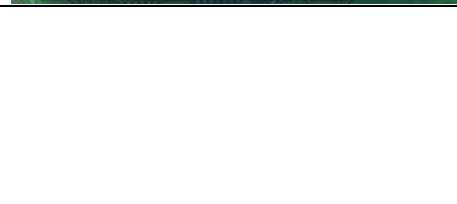
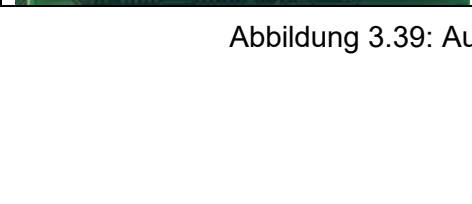
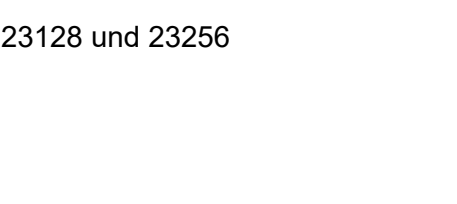
Beispiel 23128 / QL ROM		Beispiel 23256 / QL ROM
	zu identifizierende ROMS	
	← 23128 ROM 23256 ROM →	
	Auswahl des Typs	
	← Drücke SELECT Drücke SELECT →	
	← Drücke SELECT Drücke OK →	
	← Drücke OK ROM identifiziert →	
	← ROM identifiziert	

Abbildung 3.39: Auswahl der „active states“ beim 23128 und 23256

Viel komfortabler ist die automatische Erkennung der „active states“ der Chip Select und Chip Enable Signale bei einigen (P)ROMs. Wird „Autodetect“ bzw. „JUMP to detect“ im Display angezeigt, einmal JUMP drücken. Der Tester versucht jetzt zu erkennen, wie die Daten ausgelesen werden können. Folgendes zeigt die Vorgehensweise am Beispiel eines 23128:

Beispiel 23128 / QL ROM	
<pre>Selected (ROM/PROM) 23128 - 16k x 8 Vss: 14 Vtt: 28</pre>	zu identifizierendes ROM ← 23128 ROM
<pre>Select active states A: L=20,22 H=27 >Plus/4,C64,BBC,Oric SEL or OK?</pre>	Auswahl des Typs (A) ← Drücke SELECT
<pre>Select active states B: L=20,22,27 >ZX Spectrum,BBC SEL or OK?</pre>	Auswahl des Typs (B) ← Drücke SELECT
<pre>Select active states C: L=27 H=20,22 >Sinclair QL SEL or OK?</pre>	Auswahl des Typs (C) ← Drücke SELECT
<pre>Select active states D: JUMP to detect >Autodetect SEL or JUMP?</pre>	Auswahl des Typs (D) - Autodetect ← Drücke JUMP
<pre>Select active states D: L=27 H=20,22 >Autodetect SEL, JUMP or OK?</pre>	Pin 27 als low-active, Pins 20 und 22 als high-active erkannt ← Drücke OK
<pre>Calculated 23128 m9e.ic34 QL (SP) Again?</pre>	← ROM identifiziert

3.4.6 Auslesen von 2716 (2k x 8), 2532 (4k x 8), 2508 (1k x 8), 2758 (1k x 8)

Beim Auslesen von 2716 (2k x 8), 2532 (4k x 8), 2508 (1k x 8) und 2758 (1k x 8) EPROMs ist zu beachten, dass diese EPROMs eine Spannungsversorgung von 5V an Pin 21 (Vpp) auch beim Auslesen benötigen. Da der RCT an diesem Pin über keinen Vcc-Treiber verfügt, wird dieser auf ein logisch High gelegt. I.d.R. reicht das aus, um die genannten ICs auszulesen.

Es gibt aber einige EPROMs, die eine erhöhte Stromaufnahme an Vpp besitzen, so dass der Spannungspegel an Vpp zu stark abfällt. Es werden dann bei jedem Auslesen andere CRC32 Werte gelesen, d.h. der Dump ist nicht verlässlich.

Das Problem tritt auch bei anderen Testern auf und kann mit einem Zwischensockel leicht beseitigt werden.

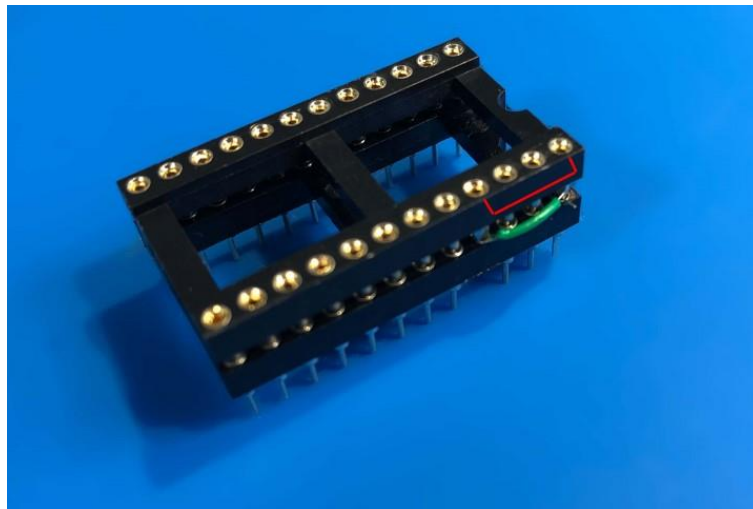


Abbildung 3.40: Zwischensockel für Vpp an 5V

An dem unteren Sockel wurde Pin 21 entfernt, damit keine Verbindung mehr zum ZIF-Sockel besteht. Die Pins 21 (Vpp) und 24 (Vcc) des oberen Sockels wurden mit einer kleinen Drahtbrücke verbunden (siehe rote Markierung).

Alternativ kann für das 2716 auch die Definition „2716 Rev (2k x 8 - EPROM).txt“ verwendet werden. Das IC muss dann 180 Grad gedreht eingesetzt werden. So kann an Vcc auch an Pin 21 abgelegt werden.

Hinweis:

Ein „TMS2516“ ist **kompatibel** zum üblichen „2716“. Ein „TMS2716“ ist **nicht kompatibel** zum „2716“, hier werden mehrere Versorgungsspannungen benötigt.



3.4.7 Auslesen eines Intel 1702A / 1602A

Es ist möglich den Intel 1702A / 1602A mit Hilfe eines Adapters auszulesen.

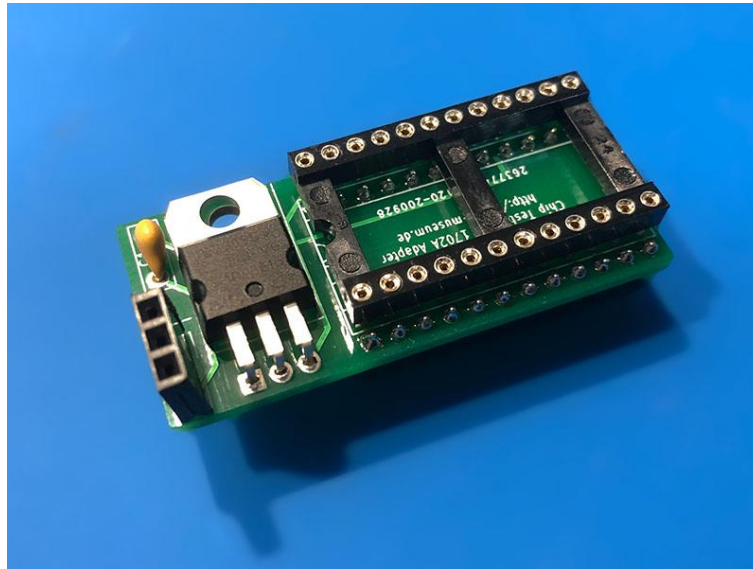


Abbildung 3.41: Adapter für Intel 1702A / 1602A

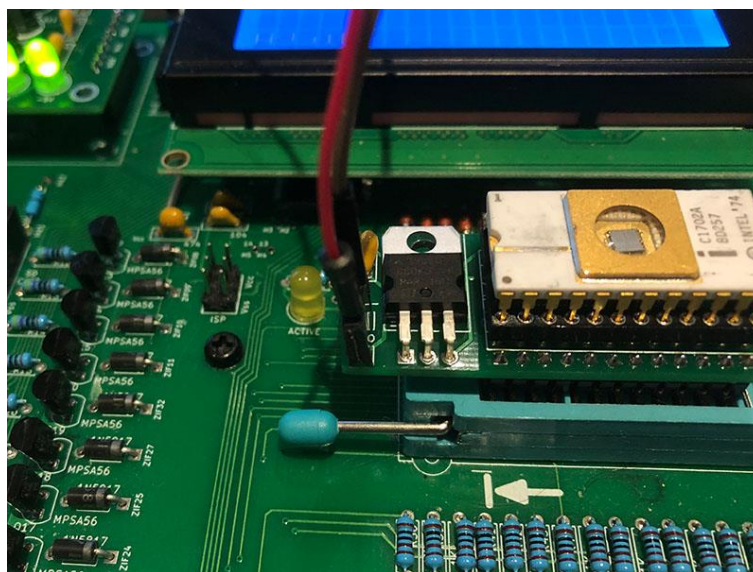


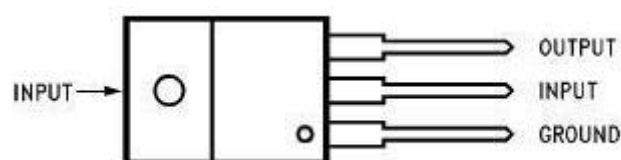
Abbildung 3.42: Adapter im Sockel

Damit der Adapter die Versorgungsspannungen für den Intel 1702A / 1602A zur Verfügung stellen kann, müssen die Anschlüsse

- „Vss/GND“ mit „Vss/GND“ am Tester (Pfostenleiste rechts neben dem DC/DC-Modul) und
- „-12V“ mit „-12V“ auf dem DC/DC-Modul

verbunden werden.

Ältere DC/DC-Modulplatinen besitzen noch keinen Anschluss für -12V, hier können die -12V am Kühlblech des LM7905 abgegriffen werden (siehe Bild, gekennzeichnet mit „Input“).



Der Adapter wurde erfolgreich mit mehreren Chips des Typs Intel 1702A und Intel 1602A getestet. Aufgrund der Beschaffenheit (und vor allem des Alters) des Chips ist aber nicht gewährleistet, dass ein Auslesen immer zuverlässig klappt.

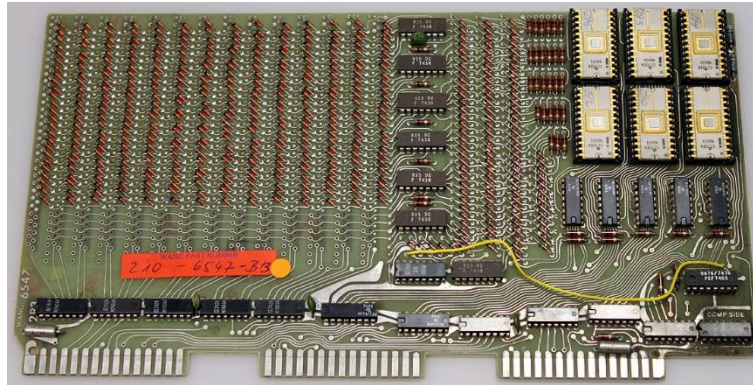


Abbildung 3.43: Getestete 1702A

Beim Auslesen bitte folgende Vorsichtsmaßnahmen einhalten, damit aus Versehen (z.B. beim Fallenlassen des Adapters) kein Kurzschluss entstehen kann:

1. Zuerst den Chip in den Adapter einsetzen.
2. Den Adapter mit Chip in den Tester einsetzen.
3. Die Spannungsversorgung mit dem Board herstellen (Vss und -12V).

3.4.8 Auslesen des Character ROM RO-3-2513 bzw. der Motorola MCM6670-Serie

Der Retro Chip Tester kann auch Character ROMs auslesen. Jedoch müssen dabei ein paar Besonderheiten dieser ROMs beachtet werden.

Character ROM RO-3-2513 (64 x 8 x 5):

Das 2513 ROM existiert in zwei Varianten: Einmal mit 5V Spannungsversorgung (mit der Bezeichnung RO-3-2513) und einmal mit +5V/-5V/-12V Spannungsversorgung. Ohne Adapter kann nur die 5V-Variante direkt ausgelesen werden.

Die Daten werden gem. der Vorgabe durch die Adressleitungen ausgelesen (Pin 14 = A0 bis Pin 22 = A8). Die fehlenden Bits 6 und 7 werden mit „0“ gepadded. Die abgelegte Datei hat eine Länge von 512 Bytes.

Character ROM MCM6670 (128 x 7 x 5):

Die MCM6670-Serie von Motorola (darunter fällt auch das SCM37530 aus dem TRS-80 Model 1) hat 7 Adressleitungen für 128 Zeichen mit 7(!) Zeilen (RS1 - RS3) und 5 Datenleitungen.

Es werden insgesamt 1024 Speicherworte mit je 5 Bit ausgelesen. Die fehlenden Bits 6 und 7 werden mit „0“ gepadded. Für die Bitkombination RS1 = RS2 = RS3 = 0 (= Zeile 0) liefert das ROM grundsätzlich „0x00“ zurück. Die abgelegte Datei hat somit eine Länge von 1024 Bytes.



Abbildung 3.44: Die ersten 21 Byte (bzw. 24 Byte mit Zeile 0) eines SCM37530

In der Abbildung wurde eine „0“ durch ein Leerzeichen und eine „1“ durch einen Stern ersetzt.

3.4.9 Auslesen des Character ROM 2513 (+5V/-5V/-12V Version)

Das Character ROM 2513 benötigt drei Spannungen: +5V, -5V und -12V. Die fehlende Spannung von -12V wird mit Hilfe eines Adapters zur Verfügung gestellt. Das Auslesen des ROMs erfolgt dann über die Einstellung 2704.

Die Daten werden gem. der Vorgabe durch die Adressleitungen ausgelesen (Pin 14 = A0 bis Pin 22 = A8). Die fehlenden Bits 6 und 7 werden mit „0“ gepadded (der Adapter besitzt hierzu Pulldown-Widerstände). Die abgelegte Datei hat eine Länge von 512 Bytes.

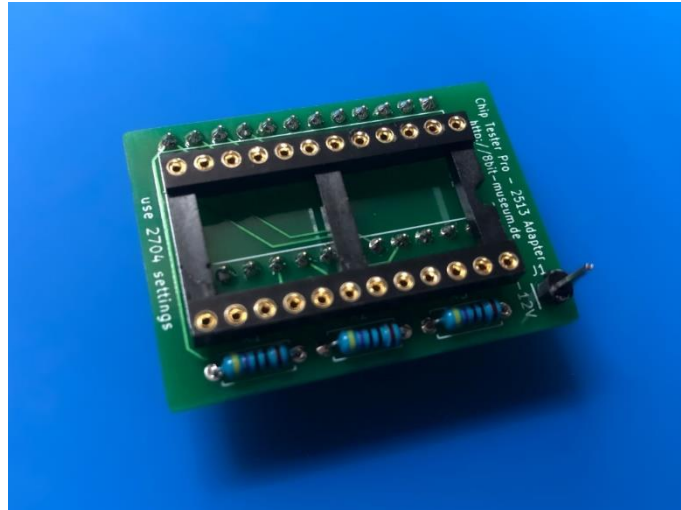


Abbildung 3.45: 2513 Adapter (+5V/-5V/12V Version)

3.4.10 Auslesen der ROMs MM4204 / MM5204

Die ROMs MM4204 und MM5204 benötigt zwei Spannungen: +5V und -12V. Die fehlende Spannung von -12V wird mit Hilfe eines Adapters zur Verfügung gestellt. Das Auslesen des ROMs erfolgt dann über die Einstellung 2704.

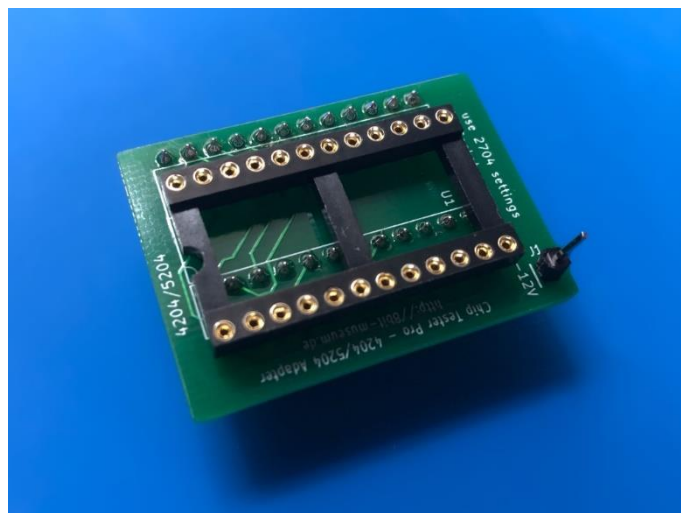


Abbildung 3.46: 4204/5204 Adapter

Achtung: Der IC ist um 180 Grad gegenüber dem Tester-Sockel gedreht.

3.4.11 Auslesen des ROM Signetics 2530

Das ROM 2530 von Signetics wurde sehr selten verwendet, z.B. im Arcade-Automaten „Qwak!“. Das ROM kann mit der benutzerdefinierten Definition „2530 (512 x 8 - ROM, ADAPTER).txt“ (ab Firmware v.22) und einem Adapter gesichert werden.

Da der Adapter recht einfach ist, kann er mit zwei 24-poligen Sockeln (runde Löcher) gebaut werden:

1. Der erste Sockel wird in den ZIF-Sockel gesteckt, der zweite auf den ersten montiert.
2. Unterer Sockel: Wir entfernen Pin 4 und Pin 12, damit gibt es keine Verbindung zum ZIF Sockel. Pin 12 wird verwendet, um 12V einzuspeisen, Pin 4 wird für Vss verwendet.
3. Oberer Sockel: Wir löten einen Draht an Pin 12 (-12V) an. Dieser Pin wird nicht mit dem unteren Sockel verbunden, sondern mit dem -12V-Anschluss auf der DC/DC-Platine.
4. Da ZIF Pin 4 kein Vss liefern kann, verwenden wir Vss von Pin 17 (ZIF Pin 25). Wir verbinden Pin 17 von der unteren Buchse mit Pin 4 der oberen Buchse.
5. Die beiden Sockel werden nun ineinandergesteckt, sodass alle anderen Pins 1:1 verbunden sind.

Für den oberen Sockel kann natürlich auch ein ZIF Sockel verwendet werden.

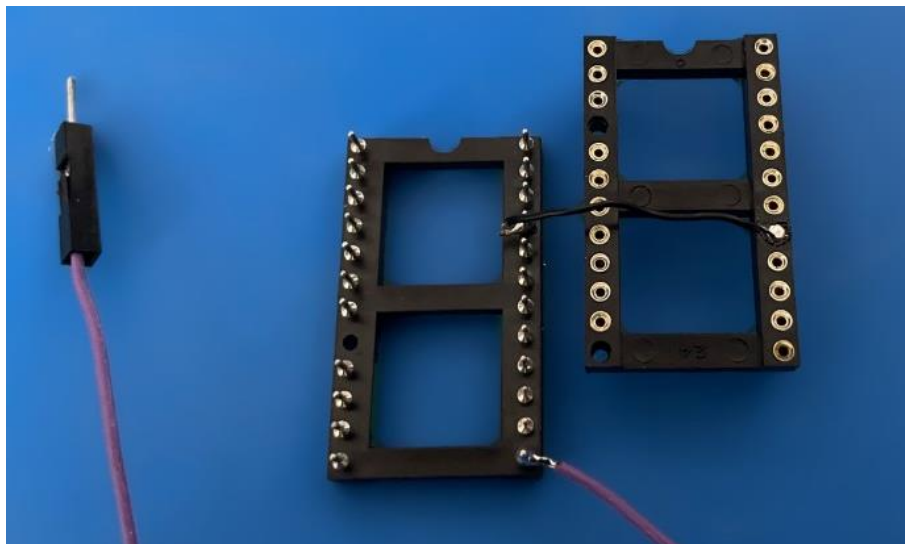


Abbildung 3.47: 2530 Adapter (Selbstbau)

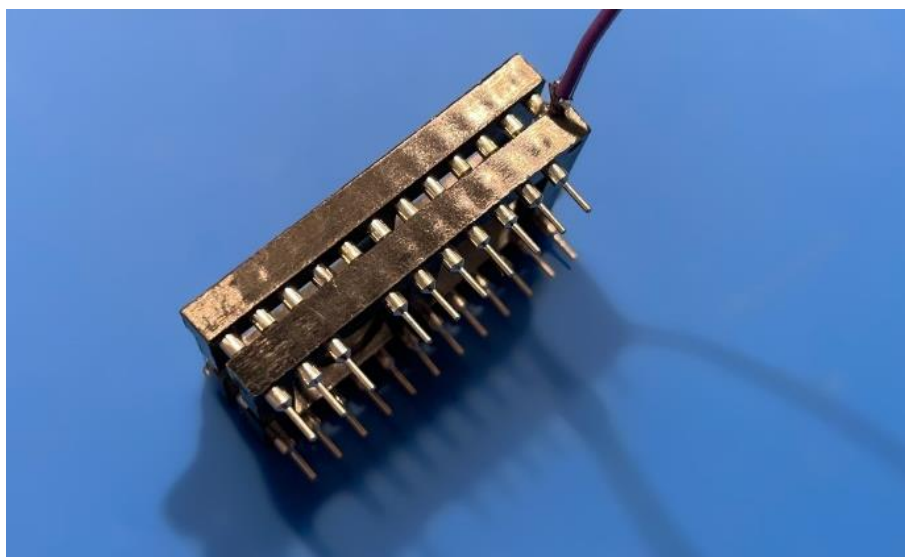


Abbildung 3.48: 2530 Adapter (Selbstbau, seitliche Ansicht)

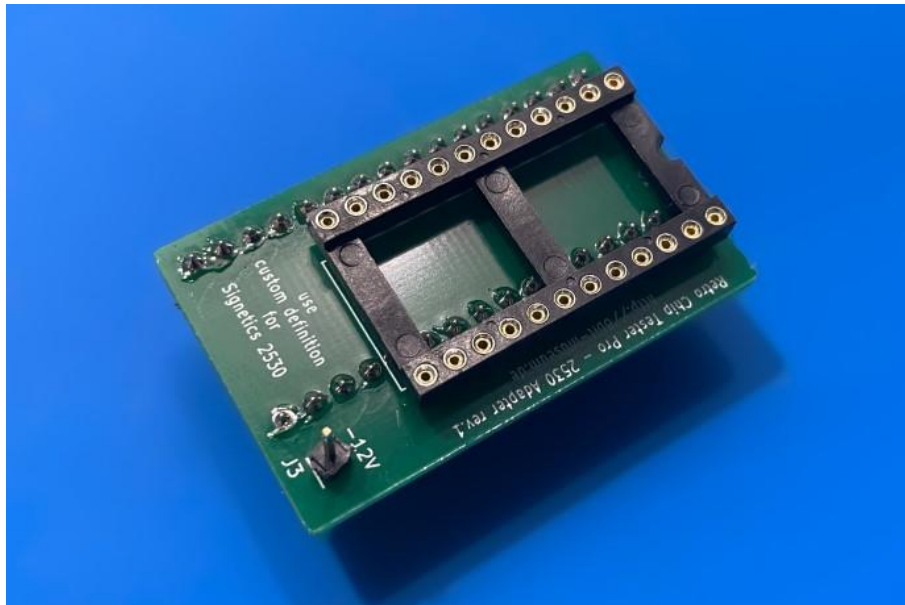


Abbildung 3.49: 2530 Adapter



Abbildung 3.50: Auslesen eines 2530 ROM

3.4.12 Auslesen des ROM Mostek MK28000 und TMS4800

Das ROM MK28000 von Mostek wurde sehr selten verwendet, z.B. im Arcade-Automaten „Tank“. Das ROM kann mit der benutzerdefinierten Definition „MK28000 – TMS4800 (2k x 8 - ROM, ADAPTER).txt“ (ab Firmware v.22) und einem Adapter gesichert werden.

Der MK28000 benötigt Pulldowns an den Datenleitungen, die durch den Adapter angelegt werden (10k Ohm). Standardmäßig arbeitet der RCT mit aktivierten Pullups bei ROMs (da viele ältere Bausteine über Open-Collector Ausgänge verfügen). Aus diesem Grund sollten die Pullups in der Konfiguration zum Auslesen eines MK28000 ausgeschaltet werden.

I.d.R. funktioniert das Auslesen zwar auch mit aktivierten Pullups (diese haben ca. 10k bis 20k Ohm), allerdings wird dann aufgrund des Spannungsteilers nur ca. 1.5v bis 2v für ein logisch LOW auf den Datenleitungen erreicht. Das sollte zwar als logisch LOW erkannt werden, verlässlicher ist aber das Auslesen mit deaktivierten Pullups. Alternativ könnten auch die Pulldowns auf dem Adapter auf 4.7k Ohm oder sogar 3.3k Ohm reduziert werden.

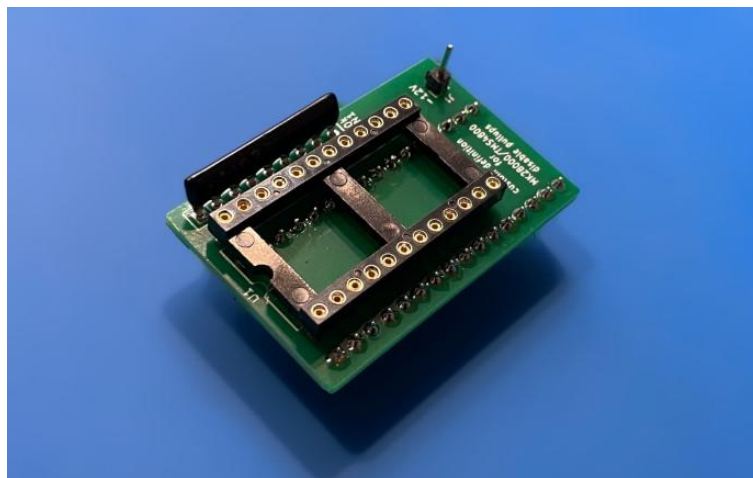


Abbildung 3.51: MK28000 Adapter

Eine selbstgebaute Version des Adapters finden Sie im BOM-Dokument.

3.4.13 Auslesen des EEPROM uPD454 und ROM uPD464

Das NEC uPD 454 und uPD 464 kann über einen einfachen Adapter ausgelesen werden. Das Auslesen erfolgt über die Custom Definition „uPD454 - uPD464 (256 x 8, EEPROM, ROM).txt“.

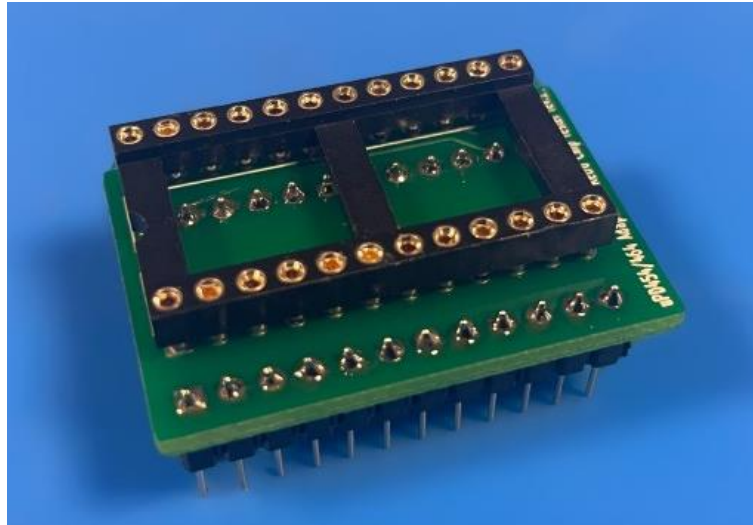


Abbildung 3.52: uPD454 / uPD464 Adapter

3.4.14 Auslesen des EPROM CDP18U42

Das EPROM CDP18U42 von RCA wurde nur selten eingesetzt. Das Auslesen des EPROMs erfolgt über die Einstellung „74S271/470“ (74S271, 74S371, 74S470, 74S471).

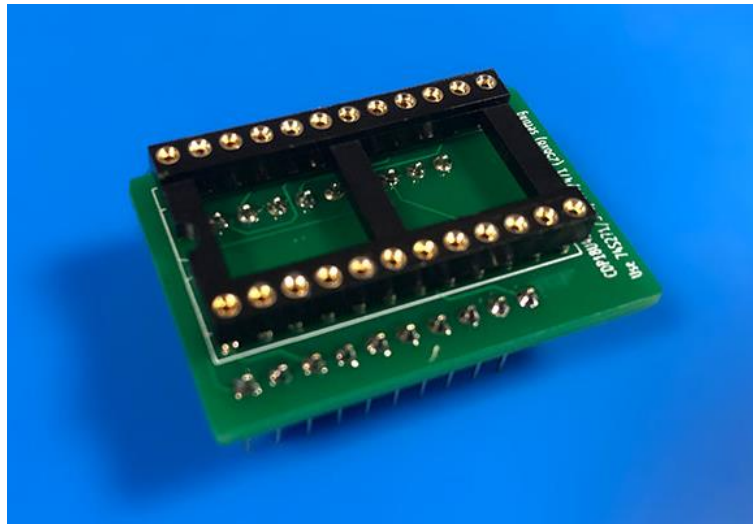


Abbildung 3.53: CDP18U42 Adapter

3.4.15 Auslesen von Commodore C64 Cartridges

Es können C64 Cartridges mit einer Speicherkapazität von 16 kByte (2x 8 kByte) ausgelesen werden. Die Umschaltung zwischen ROMH und ROML wird durch einen einfachen Schalter realisiert. Zum Auslesen wird die Einstellung 2764 (8k x 8 EPROM) verwendet.

Die Beschriftung des Cartridges zeigt beim Auslesen nach vorne (vom Display weg).

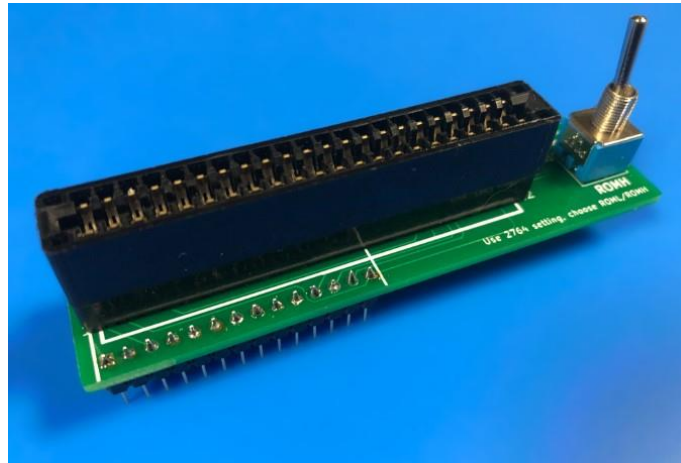


Abbildung 3.54: C64 Cartridge Adapter

Snooze Basic (1550) (Commodore) of [3]																	[27] 16d22en [2]																		
Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump	Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
00002560	00	d0	f3	4c	ba	a4	4c	b7	a4	ad	25	c5	cd	1c	c5	f0	.b0L*=L.-~\$A1.A0	00000500	00	d0	f3	4c	ba	a4	4c	b7	a4	ad	25	c5	cd	1c	c5	f0	.b0L*=L.-~\$A1.A0
00002570	09	ad	26	c5	ed	1d	c5	09	01	60	ad	26	c5	ed	1d	c5	.-\$A1.A..~-\$A1.A	00000510	09	ad	26	c5	ed	1d	c5	09	01	60	ad	26	c5	ed	1d	c5	.-\$A1.A..~-\$A1.A
00002580	60	48	49	52	45	53	40	50	4c	4f	54	40	4c	49	4e	45	HIRES\$PLOT\$LINE	00000520	60	48	49	52	45	53	40	50	4c	4f	54	40	4c	49	4e	45	HIRES\$PLOT\$LINE
00002590	40	42	4c	4f	43	4b	40	46	43	48	52	40	46	43	4f	4c	\$BLOCK\$FCHR\$FOOL	00000530	40	42	4c	4f	43	4b	40	46	43	48	52	40	46	43	4f	4c	\$BLOCK\$FCHR\$FOOL
000025a0	40	46	49	4c	4c	40	52	45	43	40	52	4f	54	40	44	52	\$FILL\$REC\$ROT\$DR	00000540	40	46	49	4c	4c	40	52	45	43	40	52	4f	54	40	44	52	\$FILL\$REC\$ROT\$DR
000025b0	41	57	40	43	48	41	52	40	48	49	20	43	4f	4c	40	49	AW\$CHAR\$HI COL\$I	00000550	41	57	40	43	48	41	52	40	48	49	20	43	4f	4c	40	49	AW\$CHAR\$HI COL\$I
000025c0	4e	56	40	46	52	41	43	40	4d	4f	56	45	40	50	4c	41	NV\$FRAC\$MOVE\$PLA	00000560	4e	56	40	46	52	41	43	40	4d	4f	56	45	40	50	4c	41	NV\$FRAC\$MOVE\$PLA
000025d0	43	45	40	55	50	42	40	55	50	57	40	4c	45	46	54	57	CE\$UPB\$UPW\$LEFTW	00000570	43	45	40	55	50	42	40	55	50	57	40	4c	45	46	54	57	CE\$UPB\$UPW\$LEFTW
000025e0	40	4c	45	46	54	42	40	44	4f	57	4e	42	40	44	4f	57	\$LEFTB\$DOWNB\$DOW	00000580	40	4c	45	46	54	42	40	44	4f	57	4e	42	40	44	4f	57	\$LEFTB\$DOWNB\$DOW
000025f0	4e	57	40	52	49	47	48	54	42	40	52	49	47	48	54	57	NW\$RIGHTB\$RIGHTW	00000590	4e	57	40	52	49	47	48	54	42	40	52	49	47	48	54	57	NW\$RIGHTB\$RIGHTW
00002600	40	4d	55	4c	54	49	40	43	4f	4c	4f	55	52	40	4d	4d	\$MULTI\$COLOUR\$MM	000005a0	40	4d	55	4c	54	49	40	43	4f	4c	4f	55	52	40	4d	4d	\$MULTI\$COLOUR\$MM
00002610	4f	42	40	42	46	4c	41	53	48	40	4d	4f	42	20	53	45	OB\$BFLASH\$MOB SE	000005b0	4f	42	40	42	46	4c	41	53	48	40	4d	4f	42	20	53	45	OB\$BFLASH\$MOB SE
00002620	54	40	4d	55	53	49	43	40	46	4c	41	53	48	40	52	45	T\$MUSIC\$FLASH\$RE	000005c0	54	40	4d	55	53	49	43	40	46	4c	41	53	48	40	52	45	T\$MUSIC\$FLASH\$RE
00002630	50	45	41	54	40	50	4c	41	59	40	3e	3e	40	43	45	4e	PEAT\$PLAY\$>>\$CEN	000005d0	50	45	41	54	40	50	4c	41	59	40	3e	3e	40	43	45	4e	PEAT\$PLAY\$>>\$CEN
00002640	54	52	45	40	45	4e	56	45	4c	4f	50	45	40	43	47	4f	TRE\$ENVELOPE\$CGO	000005e0	54	52	45	40	45	4e	56	45	4c	4f	50	45	40	43	47	4f	TRE\$ENVELOPE\$CGO
00002650	54	4f	40	57	41	56	45	40	46	45	54	43	48	40	41	54	TO\$WAVE\$FETCH\$AT	000005f0	54	4f	40	57	41	56	45	40	46	45	54	43	48	40	41	54	TO\$WAVE\$FETCH\$AT
00002660	28	40	55	4e	54	49	4c	40	3e	3e	40	3e	3e	40	55	53	(\$UNTIL\$>>\$>>\$US	00000600	28	40	55	4e	54	49	4c	40	3e	3e	40	3e	3e	40	55	53	(\$UNTIL\$>>\$>>\$US
00002670	45	40	3e	3e	40	47	4c	4f	42	41	4c	40	3e	3e	40	52	E\$>>\$GLOBAL\$>>\$R	00000610	45	40	3e	3e	40	47	4c	4f	42	41	4c	40	3e	3e	40	52	E\$>>\$GLOBAL\$>>\$R
00002680	45	53	45	54	40	50	52	4f	43	40	43	41	4c	4c	40	45	ESET\$PROC\$CALL\$E	00000620	45	53	45	54	40	50	52	4f	43	40	43	41	4c	4c	40	45	ESET\$PROC\$CALL\$E
00002690	58	45	43	40	45	4e	44	20	50	52	4f	43	40	45	58	49	XEC\$END PROC\$EXI	00000630	58	45	43	40	45	4e	44	20	50	52	4f	43	40	45	58	49	XEC\$END PROC\$EXI
000026a0	54	40	45	4e	44	20	4c	4f	4f	50	40	4f	4e	20	4b	45	T\$END LOOP\$ON KE	00000640	54	40	45	4e	44	20	4c	4f	4f	50	40	4f	4e	20	4b	45	T\$END LOOP\$ON KE
000026b0	59	40	44	49	53	41	42	4c	45	40	52	45	53	55	4d	45	Y\$DISAB\$E\$RESUME	00000650	59	40	44	49	53	41	42	4c	45	40	52	45	53	55	4d	45	Y\$DISAB\$E\$RESUME
000026c0	40	4c	4f	4f	50	40	44	45	4c	41	59	40	3e	3e	40	3e	\$LOOP\$DELAY\$>>\$>	00000660	40	4c	4f	4f	50	40	44	45	4c	41	59	40	3e	3e	40	3e	\$LOOP\$DELAY\$>>\$>
000026d0	3e	40	3e	3e	40	3e	3e	40	53	45	43	55	52	45	40	44	>\$>>\$>>\$SECURE\$D	00000670	3e	40	3e	3e	40	3e	3e	40	53	45	43	55	52	45	40	44	>\$>>\$>>\$SECURE\$D
000026e0	49	53	41	50	41	40	43	49	52	43	4c	45	40	4f	4e	20	ISAP\$CIRC\$LE\$ON	00000680	49	53	41	50	41	40	43	49	52	43	4c	45	40	4f	4e	20	ISAP\$CIRC\$LE\$ON
000026f0	45	52	4f	52	40	4e	4f	20	45	52	52	4f	52	40	4c	4c	ERROR\$NO ERROR\$B	00000690	45	52	4f	52	40	4e	4f	20	45	52	52	4f	52	40	4c	4c	ERROR\$NO ERROR\$B
00002700	4f	43	41	4c	40	52	43	4f	4d	50	40	45	4c	53	45	40	OCAL\$RCOMP\$EL\$SE\$	000006a0	4f	43	41	4c	40	52	43	4f	4d	50	40	45	4c	53	45	40	OCAL\$RCOMP\$EL\$SE\$
00002710	52	45	54	52	41	43	45	40	54	52	41	43	45	40	44	49	RETRACE\$TRACE\$DI	000006b0	52	45	54	52	41	43	45	40	54	52	41	43	45	40	44	49	RETRACE\$TRACE\$DI
00002720	52	40	50	41	47	45	40	44	55	4d	50	40	46	49	4e	44	\$PAGE\$DUMP\$FIND	000006c0	52	40	50	41	47	45	40	44	55	4d	50	40	46	49	4e	44	\$PAGE\$DUMP\$FIND
00002730	40	4f	50	54	49	4f	4e	40	41	55	54	4f	40	4f	4c	44	\$OPTION\$AUTO\$BOLD	000006d0	40	4f	50	54	49	4f	4e	40	41	55	54	4f	40	4f	4c	44	\$OPTION\$AUTO\$BOLD
00002740	40	4a	4f	59	40	4d	4f	44	40	44	49	56	40	3e	3e	40	@JOY\$MOD\$DIV\$>>\$>	000006e0	40	4a	4f	59	40	4d	4f	44	40	44	49	56	40	3e	3e	40	@JOY\$MOD\$DIV\$>>\$>
00002750	44	55	50	40	49	4e	4b	45	59	40	49	4e	53	54	40	54	DUP\$INKEY\$INST\$T	000006f0	44	55	50	40	49	4e	4b	45	59	40	49	4e	53	54	40	54	DUP\$INKEY\$INST\$T
00002760	45	53	54	40	4c	49	4e	40	45	58	4f	52	40	49	4e	53	EST\$LINE\$EXOR\$INS	00000700	45	53	54	40	4c	49	4e	40	45	58	4f	52	40	49	4e	53	EST\$LINE\$EXOR\$INS

Hex Edit View

16,480 300 100

16,480 300 100

Hex

RefEdit

8K

Abbildung 3.55: Vergleich eines Dumps mit einem Cart-Dump (CRT)

Bitte beachten, dass eine CRT einen „Header“ von 0x60 (96) Bytes enthält (nicht im Bild gezeigt). Die zu vergleichenden Informationen befinden sich ab der Startadresse 0x0060 bzw. 0x2560.

3.4.16 Auslesen von Commodore VC20 Cartridges

Es können VC20 Cartridges mit einer Speicherkapazität von 8 bis 32 kByte ausgelesen werden. Die Cartridges können beim VC20 in vier verschiedenen Speicherbereichen liegen. Diese werden mit einem Jumper ausgewählt (BLK5, BLK3, BLK2, BLK1). Fast alle Spiele, bis auf die Scott Adams Text-Adventure, liegen in Block 5 (BLK5). Bei 16K Cartridges ist gewöhnlich Block 3 (BLK3) belegt (ein paar wenige Spiele verwenden Block 1 (BLK1)). Zum Auslesen wird die Einstellung 2764 (8k x 8 EPROM) verwendet. Jeder Block muss einzeln ausgelesen werden (nur einen(!) Jumper setzen).

Die Beschriftung des Cartridges zeigt beim Auslesen nach vorne.

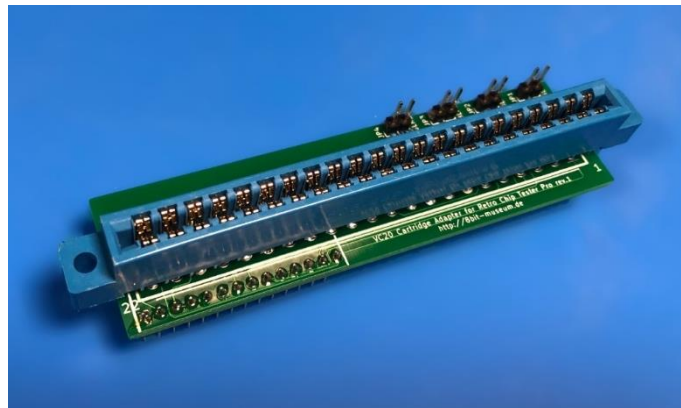


Abbildung 3.56: VC20 Cartridge Adapter



Abbildung 3.57: VC20 Cartridge Adapter

3.4.17 Auslesen von Atari VCS/2600 Cartridges

Es können VCS/2600 Cartridges mit einer Speicherkapazität von 2 kByte, 4 kByte und 8 kByte ausgelesen werden. Das Auslesen von 16 kByte Cartridges ist nicht mit jedem Cartridge möglich, da diese mit zwei ROMs möglicherweise eine hohe Stromaufnahme besitzen, die der Tester nicht bereitstellen kann. In diesem Fall zeigt jeder Dump einen anderen CRC32, d.h. es gab Fehler beim Auslesen).

Alle Cartridges (2, 4, 8, 16 kByte):

Alle Cartridges werden über den entsprechenden Menüpunkt „VCS/2600 Cart“ ausgelesen. Der Jumper **muss** bei einem 4, 8 und 16 kByte Cartridge geschlossen sein (er **darf** auch bei einem 2k Cartridge geschlossen sein). Nach Auswahl muss die Cartridge-Größe ausgewählt werden. Unterstützt wird das Standard-Bank-Switching (F8, F8SC, F6, F6SC).

2 kByte und 4 kByte Cartridges:

Der Adapter kann auch dazu verwendet werden die 2 und 4 kByte Cartridges mit einem beliebigen anderen EPROM-Lesegerät auszulesen. Hier erfolgt die Umschaltung zwischen 2k und 4k über den Jumper. Zum Auslesen wird dann die Einstellung 2716 (2k x 8) oder 2732 (4k x 8) verwendet.



Abbildung 3.58: VCS/2600 Cartridge Adapter

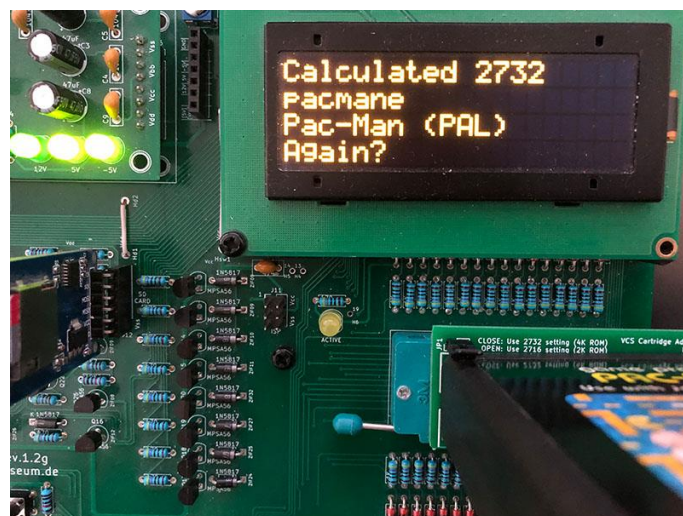


Abbildung 3.59: Ausrichtung des Cartridges

Die Beschriftung des Cartridges zeigt beim Auslesen in Richtung Display. Sollte der Adapter einen zweiten Jumper (JP2) besitzen, so ist dieser immer auf „STD“ zu setzen.

3.4.18 Auslesen von MSX Cartridges

Es können MSX Cartridges mit einer Speicherkapazität von 16 oder 32 kByte ausgelesen werden. Die korrekten Select-Signale werden über drei Jumper festgelegt: CS1 (16kb), CS2 (16kb) and CS12 (32kb).

Zum Auslesen wird die Einstellung 27128 (16k x 8 EPROM) bei 16kb und die Einstellung 27256 (32k x 8 EPROM) bei 32kb verwendet.

Die Beschriftung des Cartridges zeigt beim Auslesen nach vorne.

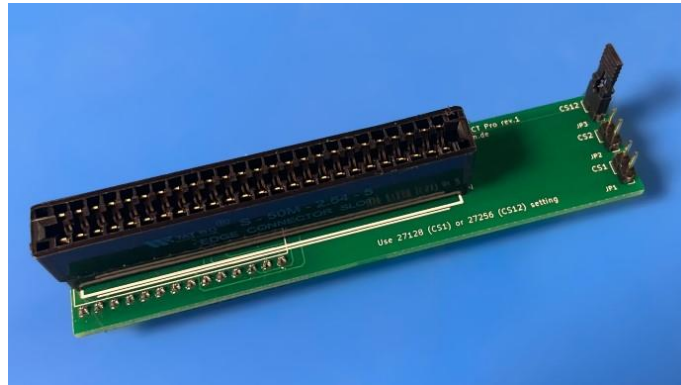


Abbildung 3.60: MSX Cartridge Adapter



Abbildung 3.61: MSX Cartridge Adapter

Hinweis: Bank Switching wird nicht unterstützt. Nur Standard 16kb und 32kb Cartridges können gelesen werden.

3.4.19 Auslesen von 1-16 Mbit EPROMs

Um 1-, 2-, 4-, 8- und 16-Mbit EPROMs zu sichern, muss der 1-16-Mbit EPROM Adapter verwendet werden. Mit diesem Adapter können EPROMs in zwei Durchgängen gesichert werden. Ein Durchgang speichert die geraden Adressen, der andere Durchgang speichert die ungeraden Adressen (JP1 wird verwendet, um den „even/odd“ (ungerade/ungerade) Modus zu wählen).



Abbildung 3.62: Orientierung des Adapters

Achtung:

- 1-, 2- und 4-Mbit-EPROMs besitzen 40 Pins. Diese EPROMs werden in dem 42-Pin-Sockel nach unten ausgerichtet (die obere Reihe bleibt frei).
- Es können nur ICs gesichert werden, die eine „even/odd“-Auswahl ermöglichen. EPROMs/ROMs mit einem festen 16-Bit Datenbus können nicht gespeichert werden.
- ROMs und PROMs funktionieren normalerweise nur in einem festen Modus (8-Bit oder 16-Bit). Nur ICs, die im 8-Bit-Modus arbeiten, können gespeichert werden.

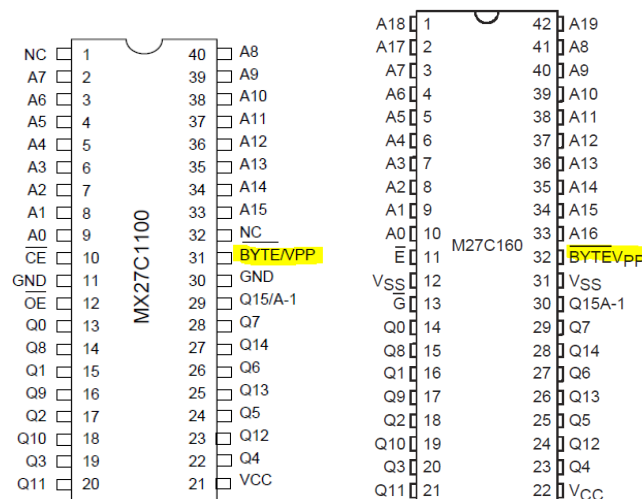


Abbildung 3.63: Unterstützte EPROMs (1- und 16Mbit Pinbelegung)

Folgende ICs mit 40 Pins sollten funktionieren:

- 1Mbit: 27C1100 (128K x 8 - 64K x 16)
- 2Mbit: 23C2000, 23C2100, 27C2100
- 4Mbit: 23C4000, 534200, 27C400, 27C4000, 27C4100

Folgende ICs mit 42 Pins sollten funktionieren:

- 8Mbit: 23C8105, 538200, 27C8192, 27C800
- 16Mbit: 23C16000, 27C160, 5316200

Bitte beachten, dass die Bezeichnung nicht durchgängig eindeutig ist. Im Zweifelsfall immer das Datenblatt konsultieren.

Um diese EPROMs zu sichern, muss zusätzlich eine benutzerdefinierte ROM-Definition installiert werden (siehe Kapitel 3.2.10). Es gibt folgende benutzerdefinierte Definitionen:

- 1MBit - 27C1100 (128K x 8 - 64K x 16 - ROM, ADAPTER).txt
- 2MBit - 23C2000 - 23C2100 - 27C2100 (256K x 8 - 128K x 16 - ROM, ADAPTER).txt
- 4MBit - 23C4000 - 534200 - 27C400 - 27C4000 - 27C4100 (512K x 8 - 256K x 16 - ROM, ADAPTER).txt
- 8MBit - 23C8105 - 538200 - 27C8192 - 27C800 (1M x 8 - 512K x 16 - ROM, ADAPTER).txt
- 16MBit - 23C16000 - 27C160 - 5316200 (2M x 8 - 1M x 16 - ROM, ADAPTER).txt

Dieser Adapter konnte noch nicht ausgiebig getestet werden. Es kann also nicht garantiert werden, dass dieser in jedem Fall funktioniert.

3.5 Testen von PLA und FPLA

Über einen kleinen Umweg können auch PLAs und FPLAs getestet werden. Die Vorgehensweise ist identisch mit dem Auslesen von ROMs (siehe Kap. 3.2.6 und Kap. 3.4).

Es wird für den zu testenden PLA eine ROM(!)-Definitionsdatei angelegt, z.B. für ein Signetics 82S100, wie es von Commodore als PLA im C64 verwendet wird, sieht diese wie folgt aus:

```
; Definition for Signetics 82S100 (64k x 8, ROM)
; Status: working, confirmed
; used to check Commodore PLAs, e.g. 906114-01, 251641-02, etc
; Rename to "customrom1.txt"
;
name: 82S100 (64k x 8)
dip: 28
size: 65536
flags: 0

a: 9,8,7,6,5,4,3,2,27,26,25,24,23,22,21,20
dout: 18,17,16,15,13,12,11,10
cs: 19
pcs:
we:
oe:

gnd: 14
vcc: 28
```

Ist die CRC32 bekannt, wird der erkannte PLA direkt angezeigt, ansonsten nur die CRC32. Diese kann dann in die externe CRC32-Datenbank aufgenommen werden.

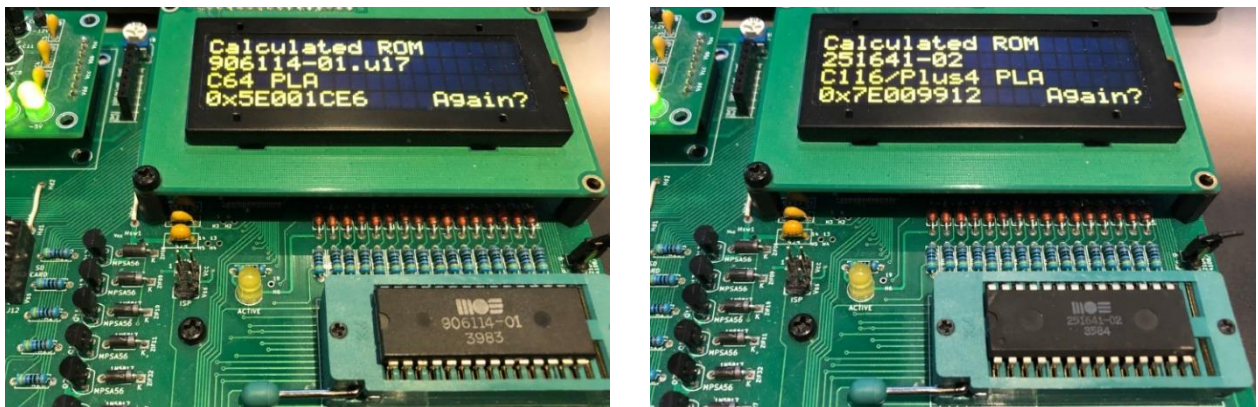


Abbildung 3.64: Identifizieren von PLAs (links C64 PLA, rechts C116/Plus4 PLA)

Es kann auch eine entsprechende Binärdatei angelegt werden mit der **versucht** werden kann den PLA mit Hilfe eines EPROMs zu ersetzen. Für den 27512 muss aber ein entsprechender Adapter verwendet werden (Eingang I_n auf Adressleitung A_n , Ausgang F_n auf Datenleitung D_n).

3.5.1 Besonderheiten bei Commodore PLAs

Es existiert bereits ein PLA2EPROM-Adapter der beim C64 (bzw. CBM-Rechnern) eingesetzt wird. Hier ist die Belegung aber nicht 1:1, wie oben erwähnt.

Das N82S100 Datenblatt zeigt dessen Pinbelegung, z.B. Pin 2 = I7, Pin 3 = I6, Pin 9 = I0, Pin 10 = F7.

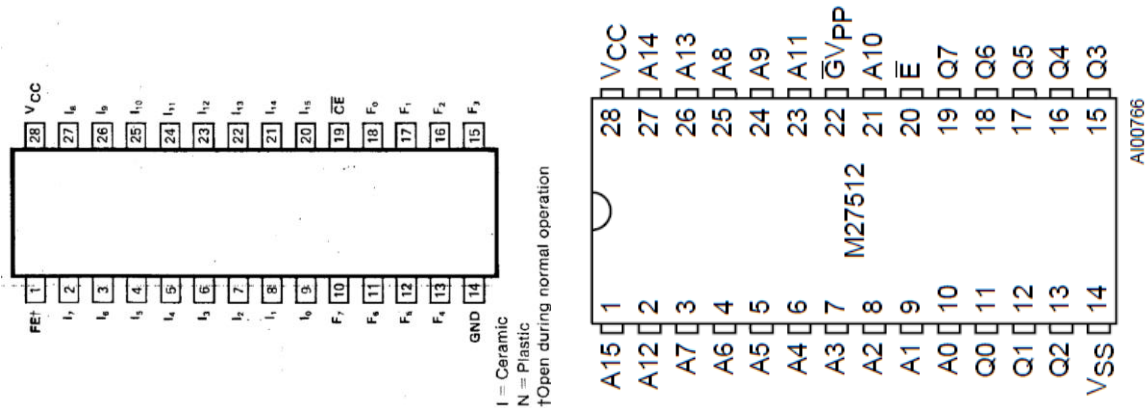


Abbildung 3.65: Pinbelegung N82S100 PLA und 27512 EPROM

Aus logischer Sicht ist es sinnvoll, diese entsprechend auf die Adress- und Datenleitungen abzubilden:

$$I_n \rightarrow A_n \text{ und } F_n \rightarrow D_n$$

Genau dieses macht die benutzerdefinierte ROM-Definition für das N82S100, wenn ein Dump erzeugt wird.

Leider ist dies unpraktisch, wenn das PLA durch ein 27512 ersetzt werden soll. Mit Fädertechnik würde die Verdrahtung zu komplex und es wäre eine Platine erforderlich. Also wurden immer, wenn ein Eingang (I) mit einer Adressleitung (A) übereinstimmte, diese verbunden und der Rest wurde verdrahtet. So war es möglich, einen Adapter mit nur zwei Sockeln zu erstellen.

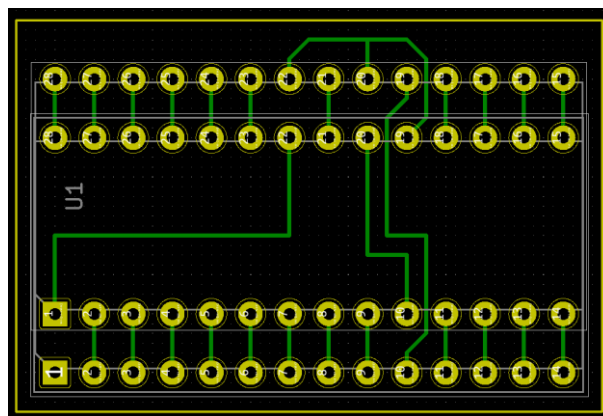


Abbildung 3.66: PLA2EPROM Adaptersockel

Das Problem dabei ist, dass die Pinbelegung für beide Chips unterschiedlich ist. Durch diesen Adapter werden z.B. I1 (Pin 8) mit A2 (Pin 8), I7 (Pin 2) mit A12 (Pin 2) verbunden. Das vereinfacht zwar die Verdrahtung, erzeugt aber das Probleme, so dass für das EPROM ein spezieller Dump genau mit diesem Adapter erzeugt werden muss.

Die PLA2EPROM Custom ROM Definition erzeugt einen Dump genau auf diese Weise, so dass er direkt für das 27512 EPROM verwendet werden kann. Bitte dennoch beachten: Das ist nur eine Quick'n'Dirty Lösung. Ein solcher Ersatz *kann* funktionieren, muss aber nicht.

Mit der folgenden ROM-Datei wird speziell für diesen Adapter ein Dump erzeugt, der in einem 27512 verwendet werden kann:

```
; Definition for PLA2EPROM dumps (64k x 8, ROM)
; Status: working, confirmed
; these dumps can be used in a 27512 as PLA replacement
; PLA-to-EPROM adapter is required
; Rename to "customrom1.txt"
;
name:  PLA2EPROM (64k x 8)
dip:    28
size:   65536
flags:  0

a:      20,9,8,7,6,5,4,3,25,24,21,23,2,26,27,22

din:    11,12,13,15,16,17,18,10
dout:   11,12,13,15,16,17,18,10
cs:     19
pcs:
we:
oe:

gnd:    14
vcc:    28
```

3.6 Lesen der „elektronischen Signatur“ eines EPROMs

Von EPROMs kann die „elektronische Signatur“ (2764-27128 und 27256-27512) ausgelesen werden.

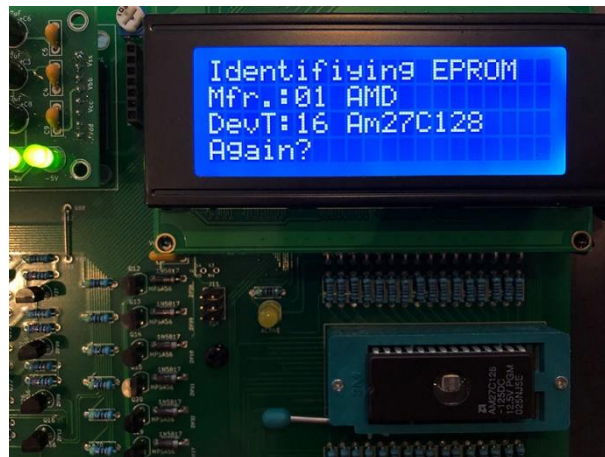


Abbildung 3.67: Identifizierung eines Am27C128 von AMD (Hersteller 0x01, Device 0x16)

Für EPROMs mit 32 Pins muss ein einfacher Adapter verwendet, da die benötigte Spannung an dem entsprechenden Pin vom RCT nicht direkt zur Verfügung gestellt werden kann:

Ein 28- und 32-poliger Sockel werden, wie in folgenden Bildern gezeigt, zusammengesteckt. Am 32-poligen Sockel werden die Pins 1, 2 und 3, sowie 29, 30 und 31 miteinander verbunden. Pin 28 vom 28-poligen Sockel wird mit Pin 32 vom 32-poligen Sockel verbunden, dieser darf keine Verbindung mit dem darüberliegenden Pin besitzen.

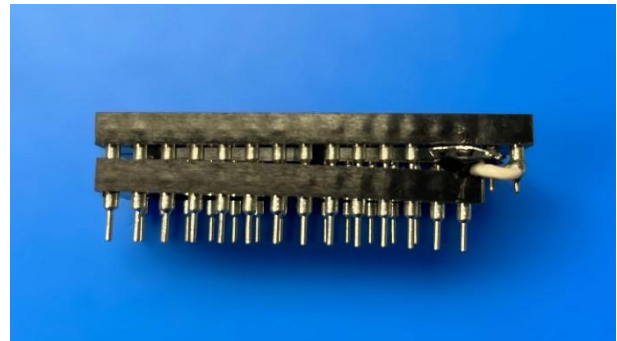


Abbildung 3.68: DIY Adapter zum Lesen der Signatur von EPROMs mit 32 Pins

Das Auslesen der Signatur erfolgt dann mit der Einstellung „ES 27256-27512“.



Abbildung 3.69: Adapter zum Lesen der Signatur von EPROMs mit 32 Pins

Beispiele:



Abbildung 3.70: Original Am29F040 (links) und „Fake“ Am29F040 (rechts)



Abbildung 3.71: Original M27C1001 (links) und Original M27C2001 (rechts)

Die Hersteller- und Typ-Signaturen sind nicht unbedingt eindeutig. Teilweise wurde dieselbe Signatur von mehreren Herstellern verwendet und dieselbe Signatur kann auch von mehreren Typen verwendet werden. Es wird bei identischen Signaturen, die häufiger verwendete angezeigt.



Es kann ebenfalls die Signatur von EEPROMs ausgelesen werden. Es ist dabei zu beachten, dass der Inhalt ggf. gelöscht werden kann.



3.7 Testen von Logik-ICs (74xx, 40xx und weitere)

Es können viele Logik-Chips der 74xx- und 40xx-Serie getestet werden. Die Logikfamilie spielt (fast) keine Rolle, da der Tester die Logik testet, keine technischen Spezifikationen, und die Spannungsversorgung 5V beträgt. Somit können aus der 74xxx-Serie die folgenden Varianten getestet werden: Standard-TTL (kein Buchstabe), L, H, S, LS, AS, ALS, F, HC, HCT, AC, ACT.⁶

Die Logik-Tester werden über das Menü aufgerufen:



Abbildung 3.72: Aufruf des Logik-Testers

Anschließend wird der zu testende Chip ausgewählt.



Abbildung 3.73: Auswahl des 7400

Sollte der Chip in Ordnung sein, wird „Chip OK“ ausgegeben, ansonsten „Chip FAILED“.

Der gefundene Fehler wird dabei für 2 Sekunden angezeigt. Die Anzeige ist wie folgt:

0 bzw. 1 = Angelegtes Signal 0V oder 5V an einem Eingang.

L bzw. H = Erwartetes Signal 0V oder 5V an einem Ausgang.

G bzw. V = Spannungsversorgung (GND bzw. Vcc)

Ein „H mit Pfeil nach unten“ bedeutet, dass ein H gelesen wurde, es wurde aber ein L erwartet.

Ein „L mit Pfeil nach oben“ bedeutet, dass ein L gelesen wurde, es wurde aber ein H erwartet.



Abbildung 3.74: Defektes „4-bit binary counter (74169)“

Im obigen Beispiel wurde ein H an Pin 11 und 13 gelesen, aber jeweils ein L wurde erwartet, und ein L wurde an Pin 12 gelesen, aber ein H wurde erwartet.

Bitte unbedingt die Hinweise in Abschnitt 3.7.4 beachten!



Ein Test wird mehrfach wiederholt, wenn er mit langem Drücken auf OK gestartet wird (siehe Konfiguration). Nicht alle Tests unterstützen dieses. FUNC (oder SELECT/JUMP) bricht vorzeitig ab.

⁶ Es gibt eine Einschränkung: Aufgrund der Strombegrenzung auf ca. 2-3mA können Bausteine mit einer besonders hohen Stromaufnahme an einem Eingang nicht unbedingt verlässlich getestet werden (i.d.R. Chips mit parallel geschalteten Flip-Flops, z.B. bei Zählern, wie 7490 (Standard-TTL)).

3.7.1 Chips mit besonderer Spannungsversorgung

Einige Chips weichen von dem Standardschema bzgl. der Spannungsversorgung ab.

Sollte hinter der Typenbezeichnung ein „Pos x“ ausgegeben werden, so muss der Chip an „Position x“ in den Sockel eingesetzt werden. Normalerweise werden die Chips nach oben ausgerichtet eingesetzt (entspricht „Pos 1“). Da einige Chips die Spannungsversorgung an nicht an den üblichen Pins (z.B. 7475, 7476, 7490, 7492, 7493) erwarten, ist dieser Chip dann versetzt einzusetzen („Pos 4“ bedeutet also, dass drei Reihen frei bleiben müssen).



Abbildung 3.75: Chip eingesetzt an Position 4 (7490)

Sollte hinter der Typenbezeichnung ein „Rev x“ ausgegeben werden, so muss der Chip an „Position x“ um 180 Grad gedreht in den Sockel eingesetzt werden. Hiervon sind aber nur sehr wenige ICs betroffen.



Abbildung 3.76: Chip um 180 Grad gedreht an Pos 1 eingesetzt (DL2416)

3.7.2 Pinbelegung von Logik-Chips

3.7.2.1 Pinbelegung von 74xx und 54xx Logik-Chips

Die Pinbelegung ist bei den unterschiedlichen Spezifikationen i.d.R. identisch:

- 54xx - Military/Airspace Grade Device
- 64xx - Industrial Device
- 74xx - Commercial Grade

Somit kann ein 5400 mit der Einstellung 7400 i.d.R. problemlos getestet werden, sofern der Gehäusotyp identisch ist. Vorsicht ist geboten, wenn z.B. ein 54xx Baustein in einem anderen Gehäuse, z.B. „W“ (ceramic flat pack), vorliegt (siehe Beispiel „7454“ unten).

3.7.2.2 Von einigen Logik-Chips gibt es unterschiedliche Ausprägungen

Der 7454 und 74LS54 haben eine unterschiedliche Funktion und damit auch Pinbelegung:

- Der 7454 beinhaltet vier AND-Gatter mit je zwei Eingängen, die mit einem NOR verknüpft sind.
- Der 74LS54 beinhaltet zwei AND-Gatter mit zwei Eingängen und zwei AND-Gatter mit drei Eingängen, die mit einem NOR verknüpft sind.

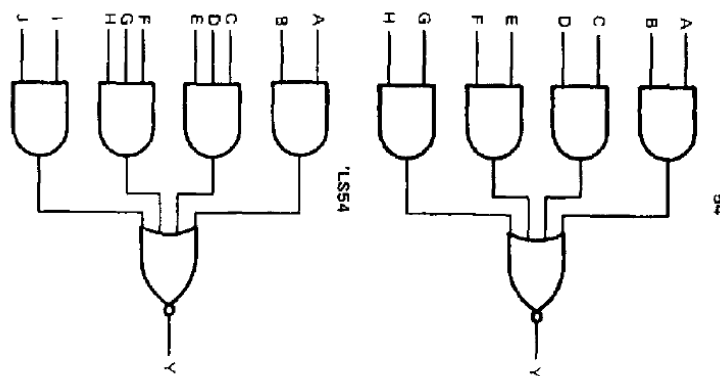


Abbildung 3.77: 74LS54 und 7454

Der Chip Tester erlaubt z.B. die Auswahl des „7454“ bzw. „7454(LS)“.

Da das Display nur eine begrenzte Anzahl von Zeichen darstellen kann, muss sollte bei älteren ICs ggf. das Datenblatt konsultiert werden. Die folgenden ICs (und ein paar weiter) werden gesondert angezeigt:

Auswahl	zu testende ICs
7451	nur 7451, H 51, S 51
7451(LS)	L 51, LS 51 und alle anderen (HC, HCT etc.), sofern vorhanden
7453	nur 7453
7453(H)	nur H 53
7454	nur 7454
7454(H)	nur H 54
7454(LS)	L 54, LS 54 und alle anderen (HC, HCT etc.), sofern vorhanden
7455(H)	nur H 55
7455(LS)	L 55, LS 55 und alle anderen (HC, HCT etc.), sofern vorhanden
7471(H)	nur H 71
7471(L)	nur L 71

Auswahl	zu testende ICs
7476	nur 76
7476(LS)	nur LS 76
7478(H)	nur H 78
7478(L)	nur L 78
7478(LS)	LS 78 und alle anderen (HC, HCT etc.), sofern vorhanden
7485	7485 und alle anderen (HC, HCT etc.), sofern vorhanden
7485(C)	nur C 85
7490	7490 und alle anderen (HC, HCT etc.), sofern vorhanden
7490(C)	nur C 90
7493	7493 und alle anderen (HC, HCT etc.), sofern vorhanden
7493(C)	nur C 93
7493(L)	nur L 93
7495	7495, LS 95 und alle anderen (HC, HCT etc.), sofern vorhanden
7495(L)	nur L 95

3.7.2.3 Unterschiedliche Pinbelegung bei verschiedenen Gehäusetypen:

Chips mit der Gehäusekennung „D“ (DIP), „J“ (CDIP = ceramic DIP) und „N“ (plastic DIP) besitzen i.d.R. dieselbe Belegung.

Chips mit der Gehäusekennung „W“ (ceramic flat pack) können u.U. eine unterschiedliche Pinbelegung haben und sollten erst nach einem Blick ins Datenblatt geprüft werden.

Beispiel 7454N/7454J und 7454W:

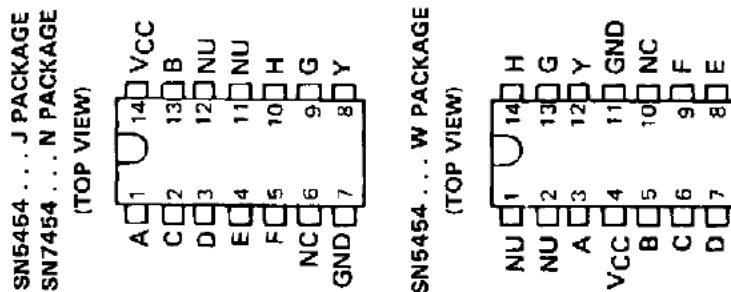


Abbildung 3.78: 7454 J/N und 7454 W

Hingegen besitzt der 74LS54 in allen Ausführungen dieselbe Pinbelegung:

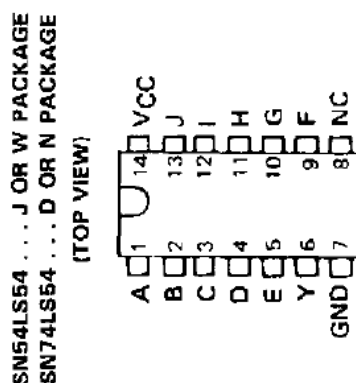


Abbildung 3.79: 74LS54 D/J/N/W

Der Chip Tester unterstützt die gängigen Pinbelegungen der DIP-Gehäusetypen.

3.7.2.4 Unterschiedliche Funktionalität aber gleiche Typennummer

Einige Logik-ICs scheinen ein bestimmter Typ zu sein, sind es aber nicht.

Zum Beispiel:

Der **SN74130** ist ein „Retriggerable Monostable Multivibrator“.

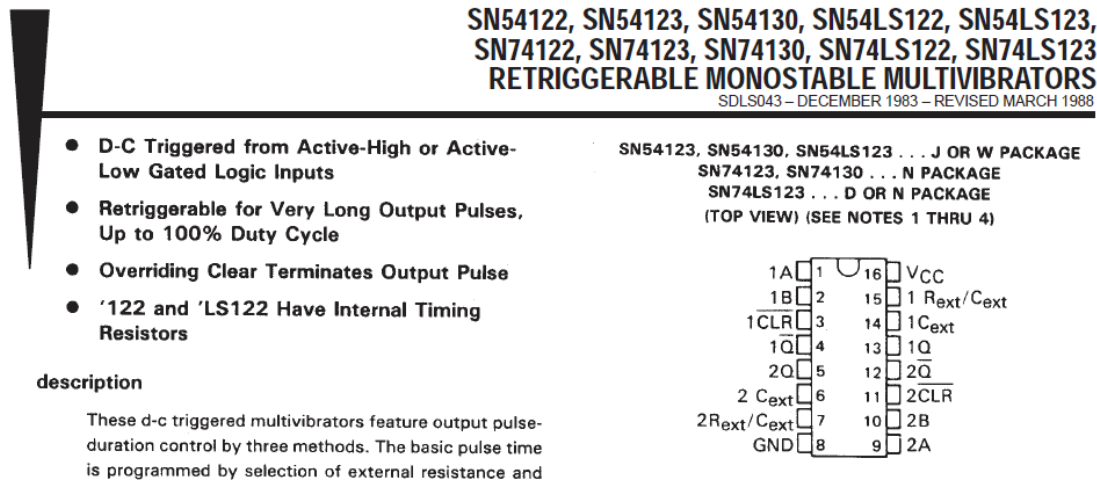
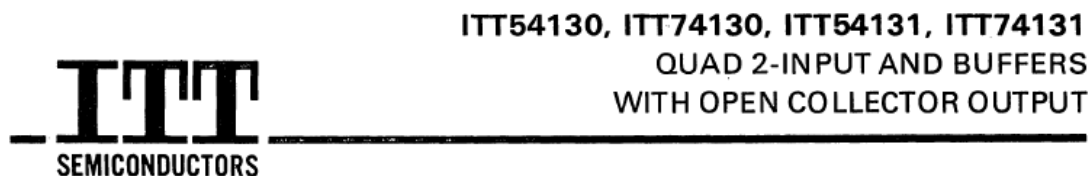


Abbildung 3.80: SN74130

Der **ITT74130** ist ein “Quad 2-Input AND Buffers with Open Collector Output”.



QUAD 2-INPUT AND BUFFERS WITH OPEN COLLECTOR OUTPUT

- High Current and High Voltage Drivers
- Inputs Are Compatible With All Other 74 Series Devices
- Large Wire-AND Capability

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage V_{CC} (see Note 1) 7V

Input voltage (see Note 1) 5.5V

Output voltage (see Notes 1 and 2):

ITT54130, ITT74130 30V

ITT54131, ITT74131 15V

Operating free-air temperature range:

ITT54130, ITT54131 -55°C to 125°C

ITT74130, ITT74131 0°C to 70°C

Storage temperature range -65°C to 150°C

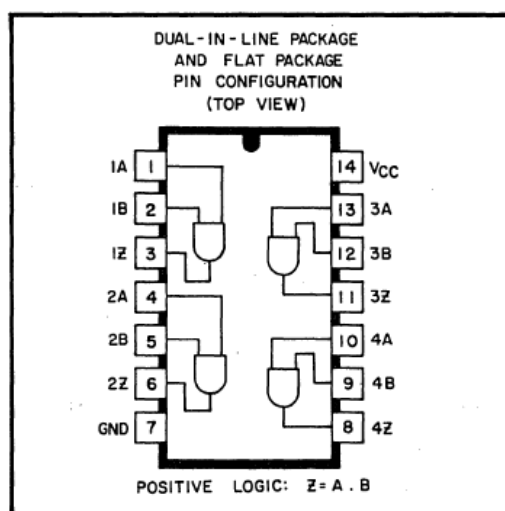


Abbildung 3.81: ITT74130

Wenn also ein Test fehlschlägt, immer zunächst prüfen, ob der IC auch die erwartete Funktion hat.

Von dem **74LS47** existieren ebenfalls funktional unterschiedliche Typen. Beim SN74LS47 von Texas Instruments wird die Ziffer „6“ ohne „Tail“ erzeugt:

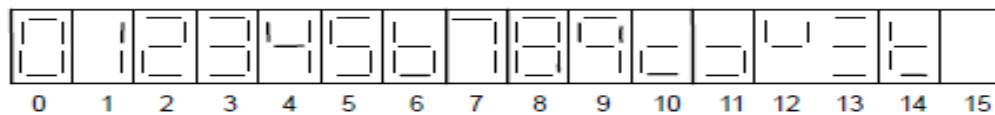


Abbildung 3.82: SN74LS47

Hingegen wird bei einem SN74LS247 die Ziffer „6“ mit „Tail“ erzeugt:

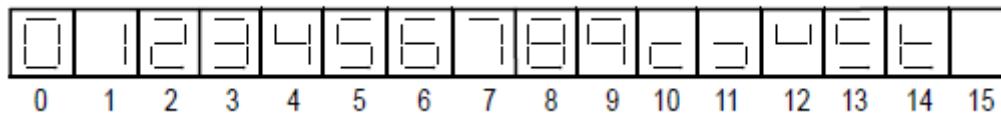


Abbildung 3.83: SN74LS247

Es gibt jedoch einige Hersteller, bei denen auch der 74LS47 ein „Tail“ bei der Ziffer „6“ ausgibt. Der Test „7447“ schlägt in diesem Fall natürlich fehl. Diese ICs müssen mit der Einstellung 74247 getestet werden.

3.7.3 Testen von russischen Chips

Russische Chips werden per Transliteration nach ISO 9:1995 angezeigt und in kyrillischer Schrift.

Schriftzeichen auf dem Chip

А	Б	к	Ц	Д	Е	Ф	г	г	И	К	Л	М	Н	л	О	П	Р	С	Т	у	В	у	х	и	ж
A	B	C	C	D	E	F	G	H	I	K	L	M	N	N	O	P	R	S	T	U	V	W	X	Y	Z

Schriftzeichen auf dem Display

Die TTL-Familien sind nach folgendem Schema gekennzeichnet:

	Soviet Union		Czechoslovakia			Poland			East Germany		
	5400	7400	5400	7400	8400	5400	6400	7400	6400	7400	8400
74	133	K155	MH54	MH74	MH84	UCA54	UCA64	UCY74		D1	E1
74L	134 136	KP134 K158									
74H	130	K131					UCA64H	UCY74H		D2	E2
74S	530	KP531	MH54S	MH74S	MH84S			UCY74S		DS	
74LS	533	K555						UCY74LS		DL...D	DL...DG
74AS	1530	KP1530									
74AL	1533	KP1533	MH54ALS	MH74ALS							
74F	1531	KP1531									
74HC	1564	KP1564									
74HCT	5564								U74HCT...DK		
74AC	1554	KP1554									
74ACT	1594	KP1594									
74LVC	5574										
74VHC	5584										

Quelle: Wikipedia

3.7.4 Testen von alten bipolaren Logik-Chips und S-TTL/H-TTL

Beim Testen von sehr alten Logik-Chips der 74xx-Serie kann es passieren, dass ein Chip als fehlerhaft getestet wird, obwohl er eigentlich korrekt arbeitet. Gleiches gilt für ICs der S-TTL und H-TTL Familie. Dieses kommt zwar sehr selten vor, es ist aber möglich, so dass dieses Problem hier kurz vorgestellt werden soll.

Der Chip Tester ist so aufgebaut, dass die I/O-Ports des ATmega2560 durch 470 Ohm Widerstände als Strombegrenzer geschützt werden (bei 5V sind somit knapp 8-9mA möglich, eher etwas weniger).

Beispiel 7425 (Dual 4-Input NOR Gates with Strobe):

Aus dem Datenblatt eines 7425 ist zu entnehmen, dass dieser an den Eingängen 1G und 2G einen Eingangsstrom von -6.4 mA für ein LOW-Signal benötigt (einfach dargestellt: in MOS-Technik sind Eingänge spannungsgesteuert, in bipolarer Technik stromgesteuert, d.h. es wird ein nicht unerheblicher Strom zum Schalten benötigt). Als Vergleich zur moderneren LS-Serie: Dort liegt ein solcher Eingang bei ca. -2mA.

PARAMETER		TEST CONDITION†	MIN	TYP‡	MAX	UNIT
I _{IL}	data inputs	V _{CC} = MAX, V _I = 0.4 V			- 1.6	mA
	strobe inputs				- 6.4	

Abbildung 3.84: Datenblatt 7425

Damit liegt der benötigte Strom knapp in dem Bereich, den der Chip Tester als Eingangsstrom liefern kann. Bei dem militärischen Varianten 5425 bzw. 8425 kann der benötigte Eingangsstrom aber noch etwas höher liegen.

Bei einem 7425 kann folgendes Problem auftreten: Der RCT testet problemlos alle Eingangskombinationen in den ersten zwei Schritten durch, im dritten Schritt wird dann folgender Fehler ausgegeben:

```

> 7425 (14)
2x 4-input NOR
110111G11011V
3 Chip FAILED Again?

```

Abbildung 3.85: Fehlermeldung 7425

Aus der Fehlermeldung ist zu entnehmen, dass der RCT die Eingänge 1G und 2G auf LOW (0) legt, an den Ausgängen aber ein „L“ gelesen wird, obwohl ein HIGH erwartet wurde. In diesem Fall reicht der Eingangsstrom nicht aus. Der Grund liegt im internen Aufbau des 7425:

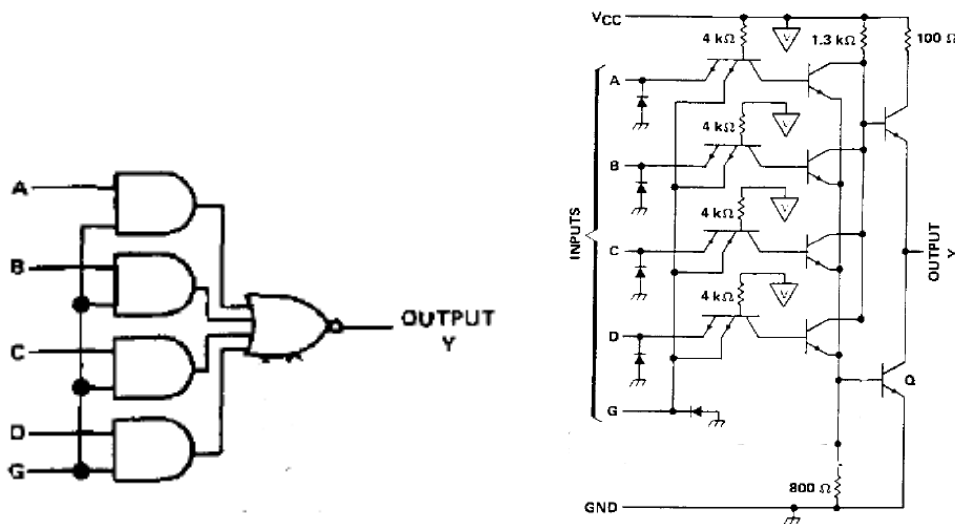


Abbildung 3.86: Logikdiagramm / Schaltbild 7425

Im Logikdiagramm des 7425 ist zu sehen, dass 1G und 2G je vier Gatter steuert und deshalb ein erhöhter Strom notwendig ist. Um einen solchen IC zu testen, müssen die Schutzwiderstände verkleinert werden. Es hat sich als Praktikabel erwiesen, die Widerstände für 1G und 1G (ZIF03 und ZIF29) temporär zu überbrücken.

I.d.R. sind in den Datenblättern Maximalwerte angegeben, d.h. der typische Wert sollte darunter liegen. Aufgrund des Alters des Chips (meistens aus den 1970ern), kann aber eine erhöhte Stromaufnahme notwendig sein. Würde man die Stromaufnahme mit der Spezifikation vergleichen, müsste der Chip vermutlich als „defekt“ aussortiert werden.

Sollen sehr alte TTL ICs aus den 1970er (kein L oder LS) getestet werden, sollte vorher das Datenblatt eingesehen werden. Sollte die Stromaufnahme eines Eingangs (meistens bei Zählern, wie beim 7490) über -6.5mA liegen, können folgende Workarounds verwendet werden:

- Einsetzen eines Pulldowns von z.B. 470 - 1k Ohm: Hierzu wird der Widerstand zwischen Vss und dem entsprechenden Eingang eingesetzt. Das kann direkt im ZIF-Sockel erfolgen.
- Temporäres überbrücken des Schutzwiderstands für den entsprechenden Eingang (Vorsicht, dass kein Kurzschluss mit benachbarten Widerständen entsteht).



Beispiel 74118 / 74119 (Hex Set-Reset Latch):

Beim 74118 und 74119 (Hex Set-Reset Latch) benötigt der gemeinsame Reset-Eingang einen hohen Eingangsstrom von knapp 10mA.

L-Eingangsstrom:

$U_S = 5,25\text{ V}$, $U_{IL} = 0,4\text{ V}$
 an $\bar{S}_1$ bis $\bar{S}_6$ und $\bar{R}_1$ bis $\bar{R}_6$
 an $\bar{R}$

$-I_{IL}$
 $-I_{IL}$

1,6 mA
 9,6 mA

Abbildung 3.87: Datenblatt 74118 / 74119

Es kann ein Pulldown von 470 Ohm zwischen R(eset) (Pin 9) und Vss (Pin 8) bzw. zwischen R(eset) (Pin 13) und Vss (Pin 12) eingesetzt werden. Die 9.6mA sind so hoch, dass auch mit der unten beschriebenen Änderung (470 Ohm Schutzwiderstände) dieser Widerstand noch notwendig sein kann.

Alternativ kann temporär auch ein Widerstand parallel zu den Schutzwiderständen geschaltet werden:



Abbildung 3.88: 1k Ohm Widerstand parallel zum 470 Ohm Schutzwiderstand

Der „Pulldown“ ist ein Workaround. Die bessere Wahl ist den Wert der Schutzwiderstände zu verringern.

**Beispiel 74H108 (altes TTL IC):**

Beim 741H108 benötigt der gemeinsame Clock-Eingang einen hohen Eingangsstrom von knapp 10mA.

Hier kann z.B. ein 470 Ohm Widerstand zwischen Masse (Pin 7) und CLK (Pin 9) gesetzt werden.

Alternativ kann temporär auch ein Widerstand parallel zu den Schutzwiderständen geschaltet werden:

Bekannte ICs mit hoher Stromaufnahme:

Bezeichnung	Beschreibung	testbar mit
7425	1G/2G = 6.4mA	470 Ohm (*)
7490	CLK2 = 4.8mA	470 Ohm
7492	CLK2 = 4.8mA	470 Ohm
7493	CLK2 = 4.8mA	470 Ohm
7494	Preset 1 / 2 = 6.4mA	470 Ohm (*)
7496	PE = 8mA	470 Ohm
74H101	CLK = 4.8 mA	470 Ohm
74H102	CLK = 4.8 mA	470 Ohm
74H103	CLK = 4.8 mA	470 Ohm
74H104	CLK = 4.8 mA	470 Ohm
74H108	CLK = 9.6 mA	< 470 Ohm
74118	R = 9.6 mA	< 470 Ohm
74119	R = 9.6 mA	< 470 Ohm
74S181	S = 8mA, C = 10mA	< 470 Ohm
74S182	G1 = 16mA, G0, G2 = 14.4mA	0 Ohm
74196	CLK2 = 6.4mA	470 Ohm (*)
DM8530	CLK2 = 4.8mA	470 Ohm
DM8532	CLK2 = 4.8mA	470 Ohm
DM8533	CLK2 = 4.8mA	470 Ohm

(*) dieser Wert ist grenzwertig, durch Alterungseffekte evtl. nicht mehr mit dem angegebenen Widerstandswert testbar

Dieses Problem haben auch andere Logiktester, z.B. der bekannte TL866 (vgl. <https://www.eevblog.com/forum/testgear/tl866-testing-ttl-7493/>), der Retro Chip Tester kann aber zumindest für diese (wenigen) problematischen Bausteine modifiziert werden.

Für Experimentier-/Risikofreudige:

Die I/O Ports des ATmega sind deshalb so hoch abgesichert, damit der Tester in jedem Fall auch Kurzschlüsse garantiert überlebt. Mit 470 Ohm an allen I/O Ports ist der Schutz i.d.R. auch noch ausreichend gut gegeben.

Wer jetzt auch noch die verbleibenden 74xx ICs testen möchte, die mehr als ca. 8mA Eingangsstrom an einem Eingang benötigen, der hat folgende Möglichkeiten:

1. Schritt: Weitere Verringerung der Schutzwiderstände:

Die I/O Ports des ATmega vertragen maximal 20mA, d.h. bei 5V darf der Schutzwiderstand rechnerisch 250 Ohm betragen (220 Ohm sind auch ok). Die Zener-Dioden vertragen maximal 500mW, bei 12V also ca. 40mA. Der Schutzwiderstand sollte für die Zenerdioden minimal also $(12-5)V / 40mA = 175 \text{ Ohm}$ betragen. Der Laststrom ist hier vernachlässigbar. Damit wäre das empfohlene Minimum 220 Ohm. Das sollte für alle alten bipolare ICs ausreichen.

Wer vorsichtig ist, kann alternativ auch alle Schutzwiderstände - bis auf die für Pin 8, 9, 27 und 28 - gegen 220 Ohm tauschen. Die Pins 8, 9, 27 und 28 können 12V führen und sollten dann bei 470 Ohm belassen werden.

2. Schritt: Entfernung der Schutzwiderstände (nicht empfohlen):

Wenn bei eigenen Experimenten die Schutzwiderstände stören, können diese (bis auf die Schutzwiderstände an Pin 8, 9, 27 und 28) entfernt werden. Das ist nicht empfohlen und sehr riskant. Sollte ein IC, der eine 12V Spannungsversorgung benötigt, einen internen Kurzschluss haben, wären die I/O Ports dieser schutzlos ausgesetzt.



3.7.5 Einsatz des Experimentier-/Breakout-Adapters

Beim 74182 (Look-Ahead Carry Generations) benötigen die gemeinsame Freigabe-Eingänge einen hohen Eingangsstrom von 14-16mA. Dieses IC ist i.d.R. nicht direkt prüfbar. Die hier vorgestellte Methode ist gleichzeitig ein Beispiel für den Einsatz des Experimentier-/Breakout-Adapters.

I_{IL} Low-level input current	C_n input	$V_{CC} = \text{MAX}, V_I = 0.4 \text{ V}$	-3.2	-3.2	mA
	$\overline{P}3$ input		-4.8	-4.8	
	$\overline{P}2$ input		-6.4	-6.4	
	$\overline{P}0, \overline{P}1, \text{ or } \overline{G}3$ input		-8	-8	
	$\overline{G}0$ or $\overline{G}2$ input		-14.4	-14.4	
	$\overline{G}1$ input		-16	-16	

Abbildung 3.89: Datenblatt 74182

Das 74LS182 kann hingegen ohne Probleme geprüft werden:

I_{IL}	C_n	$V_{IN} = 0.4 \text{ V}$ $V_{CC} = \text{MAX}$	-0.4	mA	
	$\overline{G}0, \overline{G}2$		-2.8		
	$\overline{G}3, \overline{P}0, \overline{P}1$		-1.6		
	$\overline{P}2$		-1.2		
	$\overline{P}3$		-0.8		
	$\overline{G}1$		-3.2		

Abbildung 3.90: Datenblatt 74LS182

Mit Hilfe des Breakout-Adapters kann auch dieses IC getestet werden. Hierzu wird ein 74LS07 eingesetzt, ein 6x Buffer mit Open-Collector Ausgängen, der in der Lage ist bis zu 40mA zu treiben. Die Eingänge des 74182, die mehr als 8mA benötigen werden mit dem Ausgang eines Buffers verbunden. Da der 74LS07 über Open Collector Ausgänge verfügt, werden Pull-Up Widerstände mit je 4.7k Ohm eingesetzt. Der Eingang eines Buffers wird mit dem ZIF-Sockel des RCT verbunden.

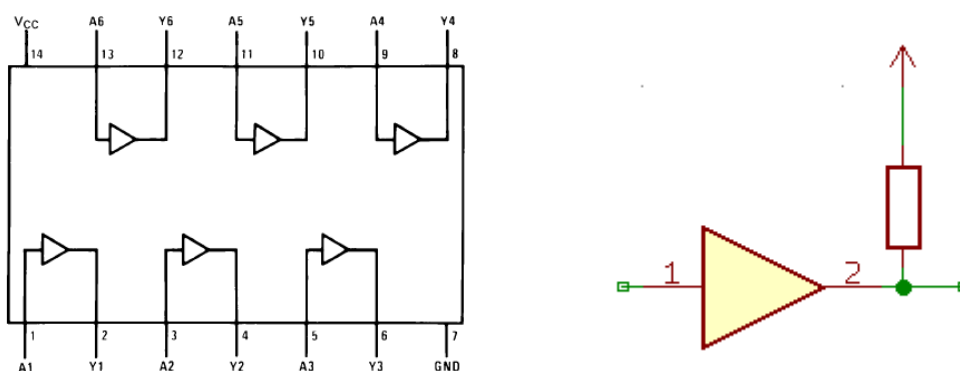


Abbildung 3.91: 74LS07 (links), Schaltung (rechts)

Der 74LS07 wird auf ein Breadboard gesetzt. Die Versorgungsspannung wird vom RCT bereitgestellt. Die Ausgänge des 74LS07 werden mit je einem Pull-Up Widerstand versehen:

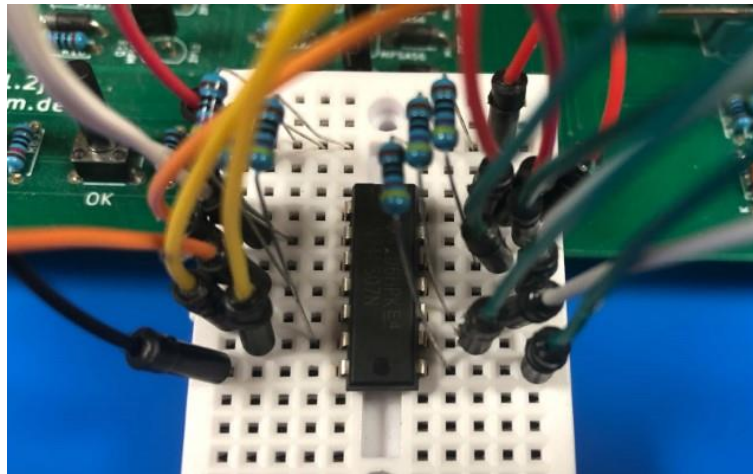


Abbildung 3.92: 74LS07 auf dem Breadboard mit Pull-Up Widerständen

Am Breakout-Adapter werden alle Pins außer Pins 1, 2, 3, 4, 5, 14 mit einem Jumper versehen.

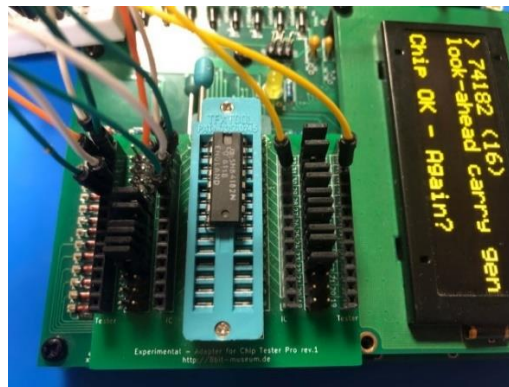
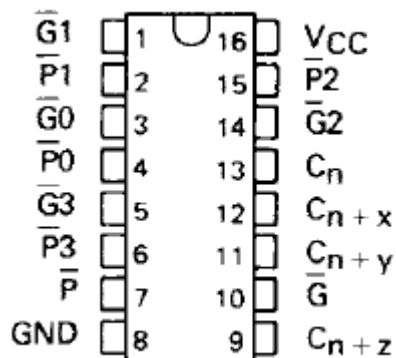


Abbildung 3.93: 74182 „Look-Ahead Carry Generations“

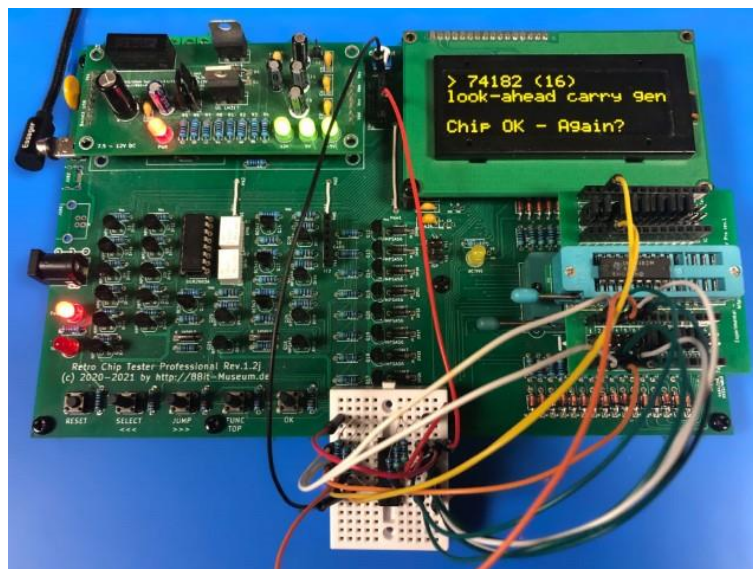


Abbildung 3.94: Vollständiger Aufbau

Die Eingänge des 74LS07 werden dann jeweils mit dem Socket-Leiste am Breakout-Adapter verbunden, die Ausgänge des 74LS07 mit der IC-Liste am Breakout-Adapter.

3.7.6 Testen von Multivibratoren (74121, 74122, 74123 et. al.)

Die Multivibratoren 74121, 74122, 74123, 74130, 74221, 74422, 74423 und 8T22 benötigen zur korrekten Funktion einige externe Komponenten (mindestens einen Widerstand und Kondensator), die durch einen Adapter zur Verfügung gestellt werden.

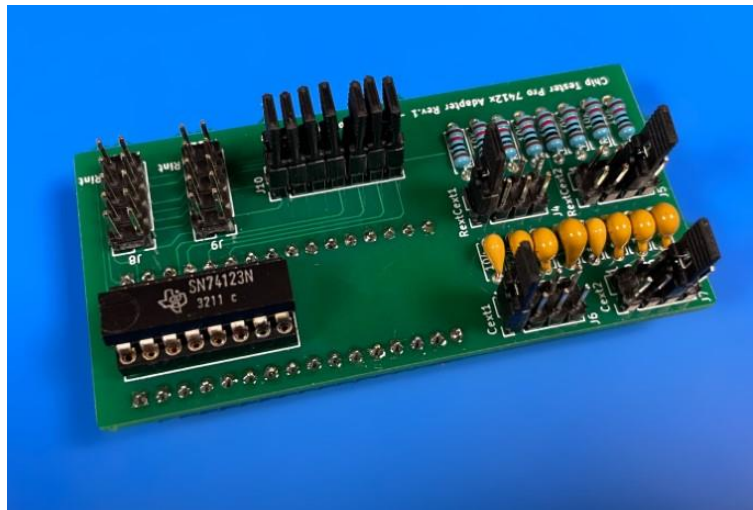


Abbildung 3.95: Adapter in 74123/74130/74221/74423 Konfiguration

Der Adapter besitzt drei Pinheader:

- 2x5 Pins für 74121,
- 2x5 Pins für 74122, 74422 and 8T22,
- 2x7 Pins für 74123, 74130, 74221 und 74423.

Zur Auswahl des zu prüfenden ICs müssen entsprechend Jumper gesetzt werden. Ein Jumper - jeweils für 74121 und 74122/74422 - ist mit „Rint“ gekennzeichnet. Dieser kann bei Bedarf gesetzt werden, wenn der interne Widerstand verwendet werden soll.

Vier Pinheader sind mit „RextCext“ und „Cext“ gekennzeichnet. Die Positionen 1 bis 4 dienen dazu eine aus vier Widerstand/Kondensator-Kombinationen auszuwählen. „RextCext2“ und „Cext2“ wird nur bei den ICs 74123, 74130, 74221 und 74423 verwendet.



Abbildung 3.96: 74123 mit gesetzter Pos 1 und Pos 3

Die Abbildung zeigt einen 74123 bei dem der erste Multivibrator in der Position 1 (10k Ohm und 100nF) und der zweite Multivibrator in der Position 3 (10k Ohm und 470nF) getestet wird.

Es wird am Ende des Tests kein „PASS“ oder „FAIL“ ausgegeben, sondern die gemessene Zeit in Microsekunden. Die folgende Tabelle zeigt, welche Werte bei den ICs ungefähr zu erwarten sind.

Typ	Konfig.		LS	HC	HCT
74121	A	1: 750 2: 1480 3: 3520 4: 6660			
74121 mit Rint	A	1: 120 2: 240 3: 580 4: 1080			
74122	B	1: 300 2: 600 3: 1160 4: 2270	1: 350 2: 690 3: 1460 4: 2850		
74122 mit Rint	B	1: 150 2: 300 3: 590 4: 1140			
74123	C	1: 270 2: 540 3: 1000 4: 1970	1: 370 2: 740 3: 1530 4: 2870		1: 450 2: 910 3: 1570 4: 3480
74130	C	?			
74221	C		1: 790 2: 1570 3: 3520 4: 6920	1: 1160 2: 2320 3: 4890 4: 10090	
74422	B		?	?	
74423	C		1: 360 2: 720 3: 1460 4: 2760	1: 440 2: 880 3: 1660 4: 3370	
8T22 (kein Rint)	B	?			

Alle Angaben in der Tabelle sind in Microsekunden. Bei den Werten handelt es sich um ungefähre (gemessene) Werte und sind abhängig von den Toleranzen der eingesetzten Widerstände und Kondensatoren. A, B, C geben die Konfiguration an.

Für die einzelnen Positionen werden folgende Werte standardmäßig verwendet:

- Position 1: 10k Ohm / 100nF
- Position 2: 10k Ohm / 220nF
- Position 3: 10k Ohm / 470nF
- Position 4: 10k Ohm / 1000nF

Bei Bedarf können auch andere Werte auf dem Adapter verwendet werden. Für das erwartete Ergebnis ist dann das entsprechende Datenblatt zu konsultieren. Die Auflösung des Timers beträgt 10us.

Wird kein Adapter benutzt, können die Komponenten auch direkt in den ZIF-Sockel gesteckt werden.

3.7.7 Testen von Multivibratoren (CD4098, CD4528, CD4538, DM9602)

Die Multivibratoren CD4098, CD4528, CD4538 und DM9602 benötigen einige externe Komponenten (mindestens einen Widerstand und Kondensator), die durch einen Adapter zur Verfügung gestellt werden.

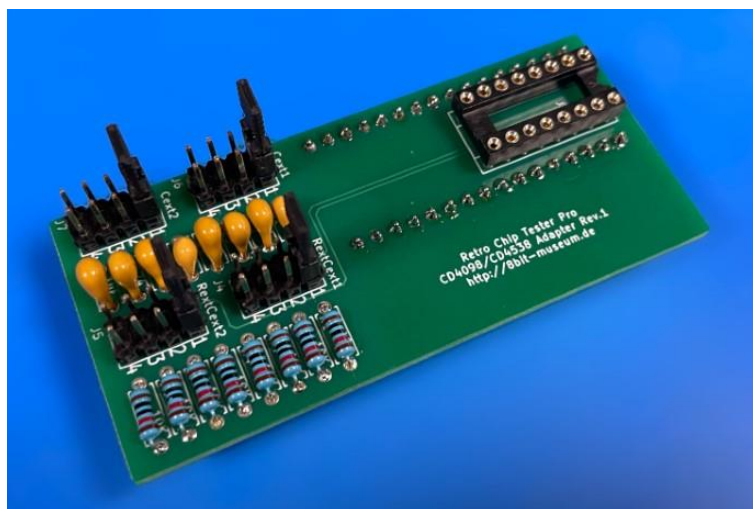


Abbildung 3.97: Adapter für CD4098, CD4528, CD4538, DM9602 mit gesetzter Pos 1

Vier Pinheader sind mit „RextCext“ und „Cext“ gekennzeichnet. Die Positionen 1 bis 4 dienen dazu eine aus vier Widerstand/Kondensator-Kombinationen auszuwählen.

Es wird am Ende des Tests kein „PASS“ oder „FAIL“ ausgegeben, sondern die gemessene Zeit in Microsekunden. Die folgende Tabelle zeigt, welche Werte bei den ICs ungefähr zu erwarten sind.

Typ	1: 100nF	2: 220nF	3: 470nF	4: 1000nF
CD4098	400	800	1150	2700
CD4528	450	950	1500	3700
MC14528	220	430	1070	2180
SCL4528	300	580	1350	2840
TC4528	310	590	1450	2910
CD4538	700	1500	2400	5800

Alle Angaben in der Tabelle sind in Microsekunden. Bei den Werten handelt es sich um ungefähre Werte und sind abhängig von den Toleranzen der eingesetzten Widerstände und Kondensatoren.

Für die einzelnen Positionen werden folgende Werte standardmäßig verwendet:

- Position 1: 10k Ohm / 100nF
- Position 2: 10k Ohm / 220nF
- Position 3: 10k Ohm / 470nF
- Position 4: 10k Ohm / 1000nF

Bei Bedarf können auch andere Werte auf dem Adapter verwendet werden. Für das erwartete Ergebnis ist dann das entsprechende Datenblatt zu konsultieren. Die Auflösung des Timers beträgt 10µs.

Wird kein Adapter benutzt, können die Komponenten auch direkt in den ZIF-Sockel gesteckt werden.

3.7.8 Testen von erweiterbaren Logik-ICs („Expander“)

Erweiterbare ICs können zwar getestet werden, da die speziellen Expander-Eingänge aber keinen TTL-Pegel führen, sind Tests nur eingeschränkt möglich. Dieses bedeutet, dass

- das Verhalten der Expander-Ausgänge nicht geprüft wird,
- ggf. die Expander-Eingänge isoliert werden müssen, damit ein Test durchläuft.

Die Expander-Ausgänge werden wie open-collector Ausgänge geprüft. Damit sollte es keine Einschränkungen bzgl. der Tests geben.

Folgende Logik-IC besitzen Expander Ein- bzw. Ausgänge:

Bezeichnung	Expander	Bemerkung
7423	Eingänge	ggf. Pins 1 und 15 isolieren
7450	Eingänge	ggf. Pins 11 und 12 isolieren
7452	Eingänge	ggf. Pin 9 isolieren
7453	Eingänge	ggf. Pins 11 und 12 isolieren
74H55	Eingänge	ggf. Pins 5 und 9 isolieren
7460	Ausgänge	Expander für 7423, 7450, 7453, 74H55
7461	Ausgänge	Expander für 7452
7462	Ausgänge	Expander für 7450, 7453, 74H55

3.7.9 Identifizieren unbekannter Logik-ICs

Der RCT kann versuchen unbekannte ICs zu identifizieren. Unter dem Menüpunkt „Identify Logic IC“ wird zunächst die Pinzahl des zu identifizierenden ICs gewählt: 14, 16, 18, 20, 24 Pins.

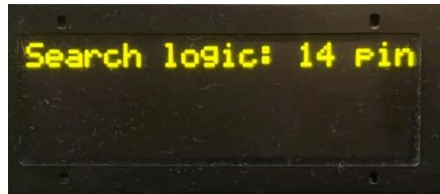


Abbildung 3.98: Identifizieren von ICs mit 14 Pins

Mit OK wird der Suchvorgang (74xx und 4xxx) gestartet, mit einem langen Druck auf OK, wird die gesamte Datenbank durchsucht (Anzeige „(all)“ im Display). Wird ein IC gefunden, kann mit OK weitergesucht werden und mit FUNC/TOP die Suche beendet werden.

Ein paar Warnhinweise:

- Bei einem IC mit N Pins wird grundsätzlich Pin (N/2) auf GND und Pin N auf Vcc kurzzeitig gelegt. ICs mit anderer Pinbelegung können dadurch Schaden erleiden.
- Defekte ICs können nicht identifiziert werden oder werden fehlerhaft identifiziert.
- Viele ICs können nicht eindeutig erkannt werden, z.B. NAND-Gatter gibt es in mehreren Ausprägungen mit identischer Pinbelegung oder ICs mit „nc“ Pins.
- ICs mit Open Collector und Tristate Ausgängen können aus technischen Gründen nicht immer eindeutig identifiziert werden.
- Es ist nicht auszuschließen, dass ICs fehlerhaft identifiziert werden, insbesondere, wenn die gesamte Datenbank durchsucht wird.
- Es wird nur versucht ICs aus der 74xx und 4xxx Serie zu identifizieren (Typen aus der DM, DS etc. Serien werden nicht versucht zu erkennen).



Warum manche ICs manchmal falsch identifiziert werden

Je mehr Ausgänge ein IC besitzt und je mehr Abhängigkeiten zu den Eingangsmustern existieren, umso einfacher ist das IC eindeutig zu identifizieren. Der RCT verwendet Kombinationen von Testmustern, um einen IC zu identifizieren. Da der RCT über 1200 ICs in seiner Datenbank verwendet, ist es durchaus möglich, dass auch ein anderes IC auf dasselbe Testmuster anspricht.

Ein konstruiertes Beispiel:

Manche ICs haben nur sehr wenige Ausgänge, z.B. haben Schieberegister manchmal nur einen einzigen Ausgang.

Ein Schieberegister könnte den folgenden Testvektor besitzen (sehr stark vereinfacht, keine Spannungsversorgung, nur ein Ausgang): 10100111H. Am Ausgang liegt ein High an, wenn an den acht Eingängen das vorausgehende Bitmuster anliegt.

Ein unbekanntes Gatter wird nun versucht zu identifizieren (in diesem Beispiel ein Triple 2-Input OR-Gatter). Das Gatter besitzt folgende Pinbelegung abyabyaby (ab = Eingänge, y = Ausgang).

Wenn der Schieberegister-Testvektor an das unbekannten IC angelegt wird, stimmt er überein: Der einzige Ausgang ist H, die beiden Bits davor sind 11 (also ist das Ergebnis der OR-Verknüpfung ein H).

Um die Erkennung konfliktfrei zu gestalten, müssen die Testpattern das IC trotz Stichprobe möglichst genau eindeutig beschreiben. Bei 1200 ICs ist das nicht trivial. Wenn ich Feedback zu fehlerhaften Identifizierungen bekomme, kann der Test ggf. verbessert werden.

3.8 Testen von sonstigen Bausteinen

Es können einige sonstige Bausteine getestet werden, wie z.B. 7-Segment Anzeigen. Diese sind im Menüpunkt „Misc Tests“ zusammengefasst.

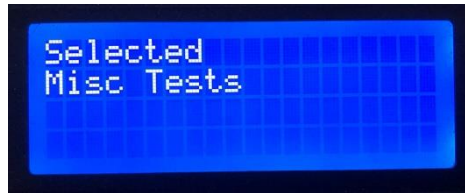
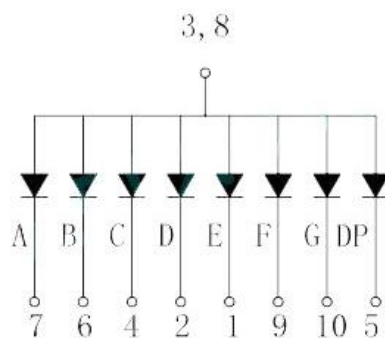


Abbildung 3.99: Aufruf der sonstigen Tests

Die Bausteine werden vergleichbar den TTL-Bausteinen in den Tester eingesetzt.

Beispiel: Ein Segment mit Punkt (DP = Dot Point), gemeinsame Anode (CA = Common Anode) an den Pins 3 und 8 als Schaltplan und mit der Anzeige im Menü:



LED 1x7 Seg., DP, CA: 3,8

Wenn unklar ist, um welchen Baustein es sich handelt, den Widerstand zwischen den Pins 3/8, 1/6 bzw. 7/9 messen (bei Anzeigen aus zwei Segmenten entsprechend). Bei 0 Ohm ist der entsprechende Baustein identifiziert.

3.8.1 Visual Fuzzy Test

LEDs, 7-Segment- und Matrix-Anzeigen können auch über den Menüpunkt „Visual Fuzzy Test“ getestet werden. Mit OK wird der Test Ein- bzw. Ausgeschaltet. Der zu testende Baustein kann an beliebiger Stelle in den Sockel eingesetzt werden.

Der Test wird mit max. 5 mA durchgeführt. Da die Spannung gepulst ist, sollten Low-Power LEDs keinen Schaden nehmen.

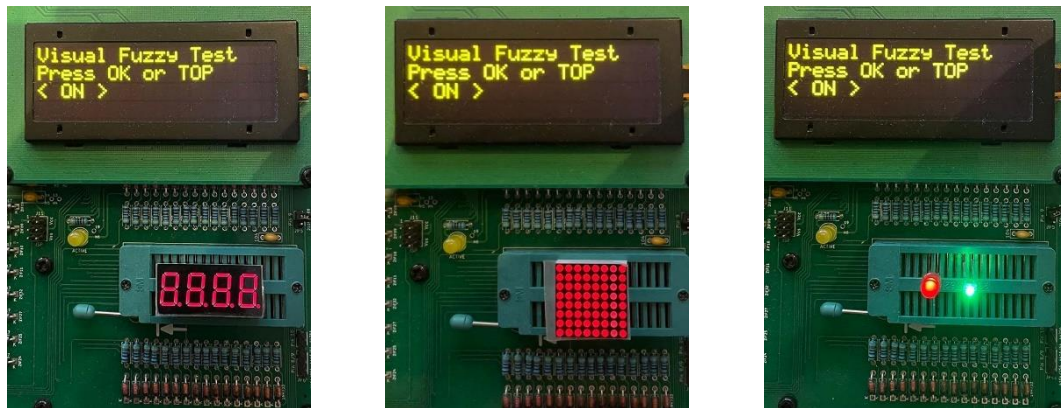


Abbildung 3.100: Visual Fuzzy Test

Bei älteren LEDs ist die Helligkeit evtl. nicht sehr hoch, da diese oft einen höheren Strom benötigen.

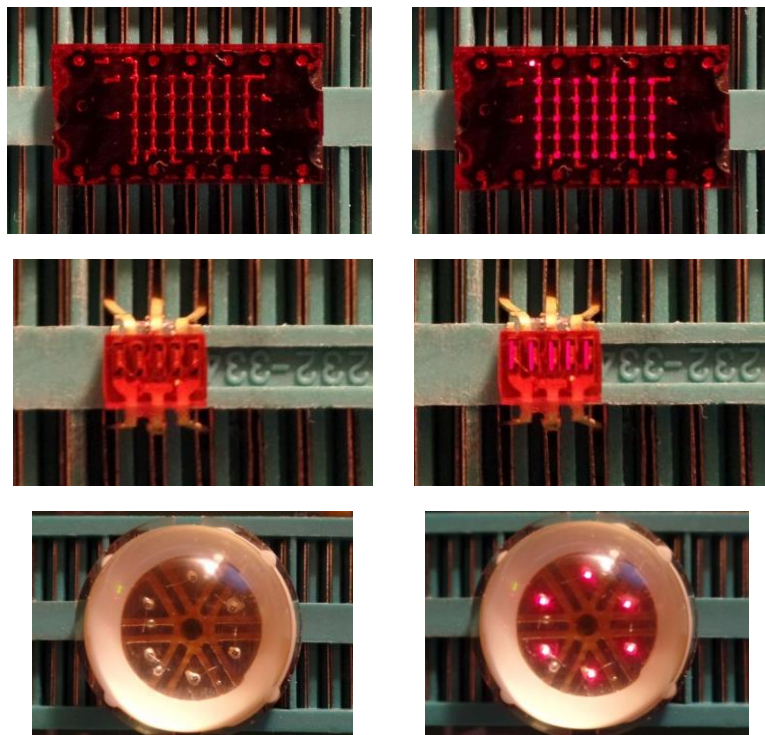


Abbildung 3.101: LEDs älterer Bauart

3.8.2 Testen von Joysticks

Mit Hilfe des Joystick Adapters kann schnell die Funktion eines Joysticks getestet werden. Es werden nur Atari-kompatible Joysticks unterstützt. Die „Auto-Fire“-Funktion kann getestet werden, sofern der Joystick nicht mehr als 10mA benötigt.



Abbildung 3.102: Joystick Adapter

Die Anzeige „UP“, „DN“, „LT“, „RT“, „FIRE“ zeigt an, ob eine Aktion erkannt wurde.

3.9 Auslesen von PALs (Programmable Array Logic)

Die 20-poligen ICs der PAL- und GAL-Familie verfügen über ein Sicherheitsbit, welches ein Kopieren des Chips verhindern soll. Die zugrundeliegenden Logikgleichungen können aber mit mehr oder weniger Aufwand rekonstruiert werden, wenn alle möglichen Eingabekombinationen durchgegangen werden und die zugehörigen Ausgaben protokolliert und anschließend analysiert werden. Auf diese Weise kann eine Kopie erstellt werden, die logisch dem Original entspricht.

Dieser Ansatz funktioniert bei folgenden ICs nicht:

- allen „registered“ PALs (PAL16R4, PAL16R6, PAL16R8 usw.)
- allen GALs, die „registered“ konfiguriert wurden (GAL16V8 usw.)

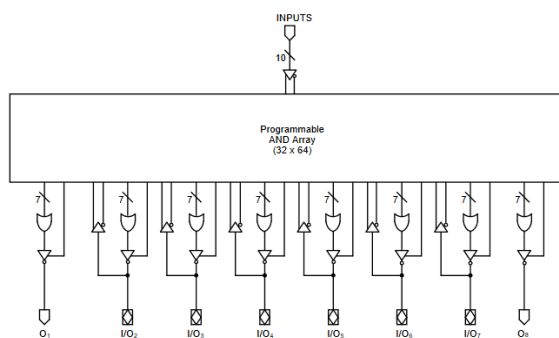
Grundsätzlich können nur reine kombinatorischen Logik-ICs so analysiert werden und keine ICs. Auch rein kombinatorische Logik-ICs, die Latches durch kombinatorische Logik implementieren, können nicht analysiert werden.

Ein GAL16V8 kann „simple“ (kombinatorisch), „registered“ oder „complex“ konfiguriert werden. Es können nur rein kombinatorische Konfigurationen ausgegeben werden. Ist Pin 11 (/OE) mit GND und Pin 1 (CLK) mit einer Taktquelle oder einem Schreib-Strobe verbunden, wird der GAL *vermutlich* „registered“ betrieben.

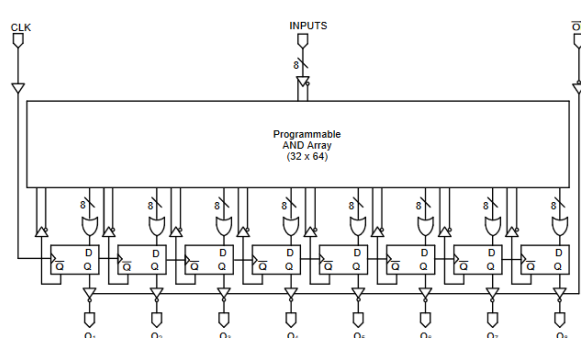
PAL Architectures Emulated by GAL16V8	GAL16V8 Global OLMC Mode
16R8 16R6 16R4 16RP8 16RP6 16RP4	Registered Registered Registered Registered Registered Registered
16L8 16H8 16P8	Complex Complex Complex
10L8 12L6 14L4 16L2 10H8 12H6 14H4 16H2 10P8 12P6 14P4 16P2	Simple Simple Simple Simple Simple Simple Simple Simple Simple Simple Simple Simple Simple

Eine weitere Einschränkung ist, dass Tri-State Ausgänge nicht erkannt werden können. Hierzu wäre zusätzliche Hardware notwendig.

Die in der Tabelle rechts als „simple“ aufgeführten PALs bzw. durch einen GAL16V8 emulierten Bausteine können kombinatorisch ausgelesen werden. Die mit „complex“ bezeichneten Bausteine können u.U. ausgelesen werden.



PAL16L8 (kombinatorisch)



PAL16R8 („registered“)

3.9.1 Vorbereitung

Diese Funktion ist nur zusammen mit einer SD-Karte, welche die gelesenen Werte in Form einer Wahrheitstabelle aufnimmt, sinnvoll einsetzbar.

Hierzu muss in der Konfiguration die Unterstützung für SD-Karten eingeschaltet werden und der Micro-SD Kartenadapter muss mit dem Chip-Tester verbunden werden. Der Anschluss ist in Kap. 5 beschrieben.

Das Auslesen eines PALs wird über den folgenden Menüpunkt (bzw. „PAL 24pin“) ermöglicht:

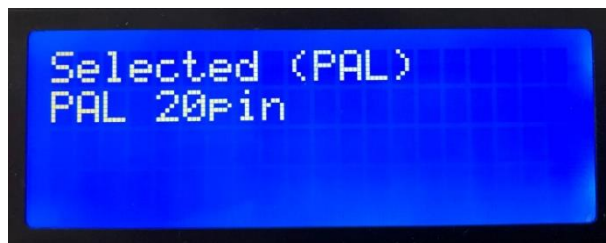


Abbildung 3.103: Menü „PAL“

Die Verwendung dieses Features ist ähnlich dem Auslesen eines ROMs oder EPROMs, d.h. wird bei eingeschalteter Unterstützung:

... **die OK Taste lang gedrückt**, so wird der Inhalt des PALs in eine Datei geschrieben.

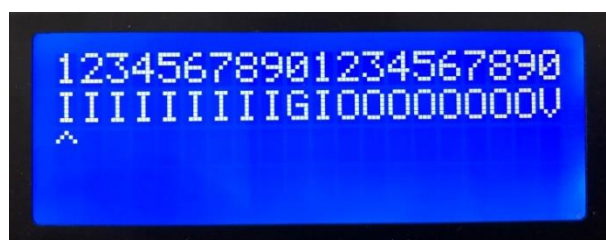
... **die OK Taste kurz gedrückt**, wird über die ermittelten Ausgangswerte nur eine CRC32 berechnet.

Der CRC32-Wert ist z.Zt. wertlos, er wird derzeit benutzt, um der Ausgabedatei auf der SD-Karte einen eindeutigen Namen zu geben. Er kann ggf. dazu verwendet werden, zwei Bausteine miteinander zu vergleichen.

Im nächsten Schritt muss der Baustein definiert werden, es muss festgelegt werden, welche Pins zur Eingabe- und welche Pins zur Ausgabe verwendet werden.



Einstellung („OK“ kurz gedrückt)



Einstellung („OK“ lang gedrückt)

Voreingestellt ist ein PAL10H8 bzw. PAL10L8, der an den Pins 1 bis 9 und 11 über Eingänge und an den Pins 12 bis 19 über Ausgänge verfügt. GND liegt an Pin 10 und Vss an Pin 20 an.

I0	1	20	Vcc
I1	2	19	O7
I2	3	18	O6
I3	4	17	O5
I4	5	16	O4
I5	6	15	O3
I6	7	14	O2
I7	8	13	O1
I8	9	12	O0
GND	10	11	I9

Abbildung 3.104: Vorbelegter PAL10H8 bzw. PAL10L8

Der Pin mit der niedrigeren Nummer ist dabei dem niederwertigeren Bit zugeordnet.

Die Belegung kann über das Tastenfeld geändert werden. Durch Druck von SELECT und JUMP gleichzeitig wird zwischen den zwei Gruppen [IO-] und [GV] umgeschaltet. Durch Drücken von JUMP wird dann innerhalb der gewählten Gruppe durchgeschaltet, also [I] = Eingabe → [O] = Ausgabe → [-] = Ignoriert bzw. [G] = GND → [V] = Vss. Mit SELECT wird der nächste Pin ausgewählt. OK startet den Auslesevorgang.

Achtung:

Es werden maximal 8 Ausgabe-Pins unterstützt. Sind mehr angewählt werden nur die unteren 8 Bit (also die ersten acht als Ausgabe definierten Pins) gespeichert!



Werden mehr als acht Masse bzw. Vcc definiert, werden nur die ersten acht Definitionen beachtet.



Die zuletzt eingestellte Belegung bleibt bis zum nächsten RESET erhalten.



3.9.2 Starten des Auslesevorgangs

Wird der Auslesevorgang gestartet, wird zunächst für 5 Sekunden die Konfiguration angezeigt:

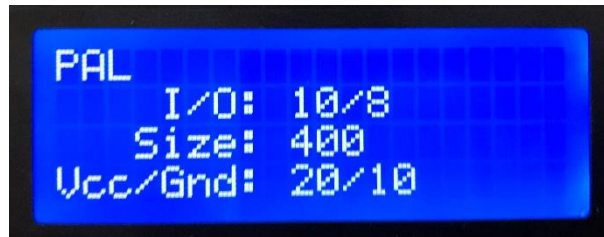


Abbildung 3.105: PAL10H8/PAL10L8: 10 Eingänge, 8 Ausgänge, 1024 Eingangskombinationen
Vcc = Pin 20 und GND = Pin 10

Danach wird der Auslesevorgang gestartet und am Ende eine CRC32 angezeigt, die über die ermittelten Ausgabewerte berechnet wurde. Dieser Wert wird als Dateiname für eine Binärdatei verwendet, welche die Ausgabewerte (maximal 8) speichert.

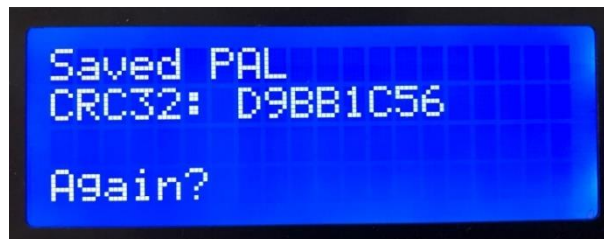


Abbildung 3.106: PAL mit CRC32

Weiterhin wird eine CSV-Datei gespeichert, die alle Eingabewerte mit den dazugehörigen Ausgabewerten beinhaltet. Die erste Zeile beinhaltet die Pin-Nummern bezogen auf einen 20- oder 24-poligen IC.

Je nach Anzahl der Eingänge kann der Speichervorgang einige Minuten dauern.

Mehr Informationen dazu in Kapitel 8.

3.9.3 Weitere Möglichkeiten für den Einsatz mit Beispiel

Die PAL-Auslesefunktion kann grundsätzlich für alle Chips verwendet werden, die rein kombinatorisch arbeiten. So kann z.B. auch das Verhalten eines normalen 7400 damit aufgezeichnet werden. Hierzu müssen die Ein- und Ausgänge des Chips vorgegeben werden. Da der 7400 nur über 14 Pins verfügt, sind die Pins 8 bis 13 in dem „virtuellen 20-poligen Sockel“ nicht belegt und müssen ignoriert werden. Die Spannungsversorgung ist an Pin 7 (GND) und Pin 14 (Vss) des ICs bzw. Pin 20 des Sockels.



Abbildung 3.107: Definition eines 7400 (4-fach NAND)

Im nächsten Schritt bekommen wir die Konfiguration angezeigt:

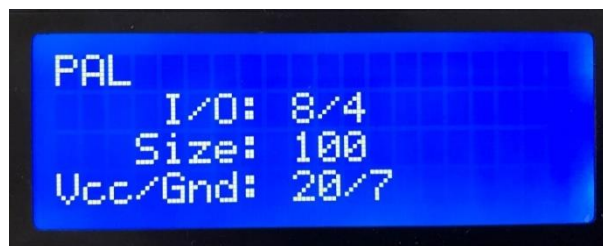


Abbildung 3.108: 7400 (4-fach NAND): 8 Eingänge und 4 Ausgänge, 256 Eingangskombinationen
Vcc = Pin 20 und GND = Pin 7 (Achtung: Zählung bei einem 20-poligen Sockel)

Nach dem Auslesen befinden sich zwei Dateien auf der SD-Karte.

Eine Binärdatei, die nur die Ausgabewerte beinhaltet (da vier Ausgänge vorhanden sind, sind diese in den unteren 4 Bits gespeichert):

```

00000000 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
00000010 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
00000020 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
00000030 0b 0b 0b 0a 0b 0b 0b 0a 0b 0b 0b 0a 09 09 09 08
00000040 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
00000050 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
00000060 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
00000070 0b 0b 0b 0a 0b 0b 0b 0a 0b 0b 0b 0a 09 09 09 08
00000080 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
00000090 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
000000a0 0f 0f 0f 0e 0f 0f 0f 0e 0f 0f 0f 0e 0d 0d 0d 0c
000000b0 0b 0b 0b 0a 0b 0b 0b 0a 0b 0b 0b 0a 09 09 09 08
000000c0 07 07 07 06 07 07 07 06 07 07 07 06 05 05 05 04
000000d0 07 07 07 06 07 07 07 06 07 07 07 06 05 05 05 04
000000e0 07 07 07 06 07 07 07 06 07 07 07 06 05 05 05 04
000000f0 03 03 03 02 03 03 03 02 03 03 03 02 01 01 01 00

```

Abbildung 3.109: Datei: d9bb1c56.bin

Und eine CSV-Datei, die sowohl die Ein- als auch die Ausgabewerte in lesbarer Form beinhaltet (hier nur die ersten 20 Werte von insgesamt 256 Werten):

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

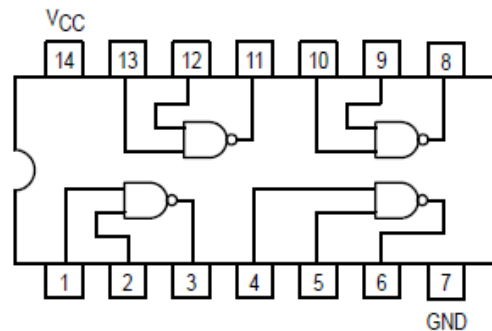


Abbildung 3.110: Wahrheitstabelle

Die erste Zeile (in grau) beinhaltet die Pin-Nummern bezogen auf einem 20-poligen Sockel. Die gelbe Zeile wurde zu Verständnisgründen hinzugefügt und beinhaltet die korrigierten Pin-Nummern bezogen auf einen 14-poligen Sockel.

Sehr schön ist das Verhalten der Ausgänge an Pin 3 und Pin 6 zu erkennen, die immer dann LOW sind, wenn beide Eingänge, also Pin 1/2 bzw. Pin 4/5, auf HIGH sind (NAND-Gatter).

Die Binärdatei enthält die Werte der rechten Spalte:

- 1. Byte den Wert 0x0f (1111)
- 2. Byte den Wert 0x0f (1111)
- 3. Byte den Wert 0x0f (1111)
- 4. Byte den Wert 0x0e (1110)
- usw. bis zum 256. Byte

Nachdem die Wahrheitstabelle erzeugt wurde, beginnt der u.U. knifflige Teil: Die Schaltungssynthese, in der aus der Wahrheitstabelle die Funktionsgleichung als disjunktive oder konjunktive Normalform erstellt wird. Ein wesentlicher Bestandteil ist die Optimierung und das Finden von minimierten Funktionsgleichungen mit den De Morganschen Gesetzen und der Schaltalgebra. Auch KV-Diagramme können bei der Optimierung helfen, zumindest wenn wenige (bis zu vier) Eingabevariablen vorhanden sind.

Weitere Links:

- https://de.wikipedia.org/wiki/Boolesche_Funktion
- https://de.wikipedia.org/wiki/Disjunktive_Normalform
- <https://de.wikipedia.org/wiki/Vollkonjunktion>
- <https://de.wikipedia.org/wiki/Karnaugh-Veitch-Diagramm>

3.9.4 Alternative Möglichkeit PALs und GALs zu rekonstruieren

RCT-HQ gibt keinen Support für die folgend genannte Software!

Eine weitere Methode die Logikgleichungen eines PALs bzw. GALs zu rekonstruieren ist auf der Website

<http://dreamjam.co.uk/emuviews/readpal.php>

beschrieben.

Der Adapter ist sehr simpel. Im Grunde werden 8 Adressleitungen als Pullup/Pulldown für die Datenleitungen verwendet, um den Zustand der I/O Pins zu ermitteln.

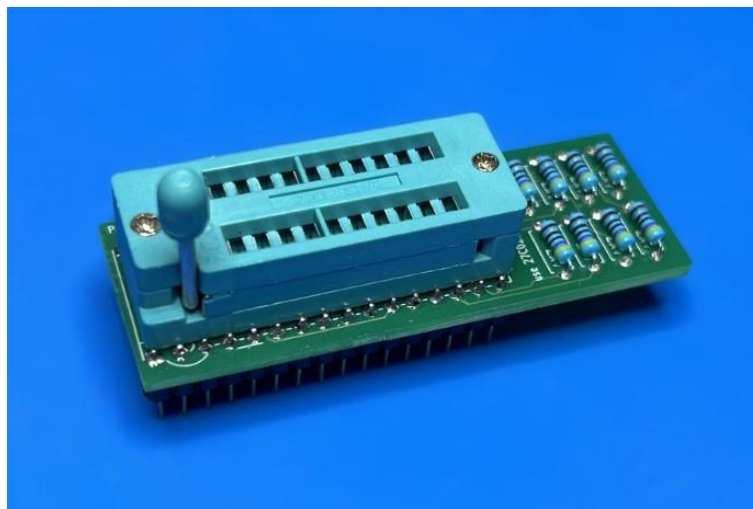


Abbildung 3.111: SecurePAL16v8 Adapter

Die auf der Website veröffentlichte Software ist schon recht alt, sie ist aber funktionsfähig. d.h. man kann ein PAL16x8 bzw. GAL16x8 mit dem RCT "dumpen" und mit der Software eine PLD-Datei erzeugen, die mit WinCUPL minimiert werden kann (es gelten die gleichen Einschränkungen wie bei der oben bereits beschriebenen RCT Methode).

Die Vorgehensweise ist grundsätzlich wie folgt:

1. Mit Hilfe des Adapters den PAL als 272001 (im Menü EPROM) speichern.
2. Die Software von o.g. Website erzeugt aus der BIN-Datei eine PLD-Datei für WinCUPL.
3. WinCUPL minimiert die Terme und erzeugt eine JED-Datei mit der ein PAL/GAL programmiert werden kann.

Download-Link für WinCUPL: <https://www.microchip.com/en-us/development-tool/wincupl>

Die Gerber-Dateien für den Adapter sind auf der RCT-Firmware Seite verfügbar.

3.10 Testen von SOIC (SOP) Bausteinen

Mit Hilfe eines Adapters können grundsätzlich auch ICs in den Bauformen SOIC bzw. SOP getestet werden. Diese Adaptersockel sind für wenige Euro auf gängigen Marktplätzen erhältlich.

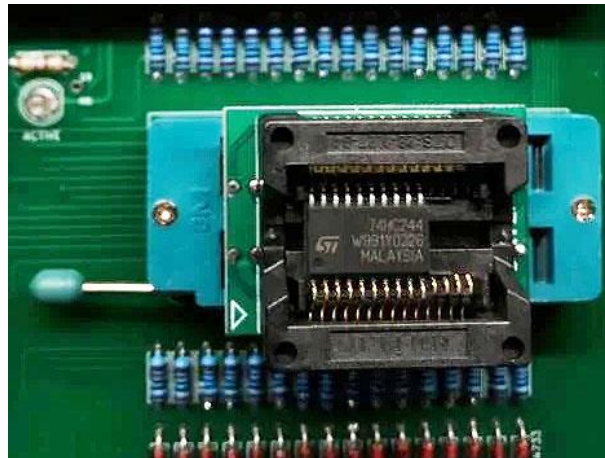


Abbildung 3.112: SOIC-28 Adapter

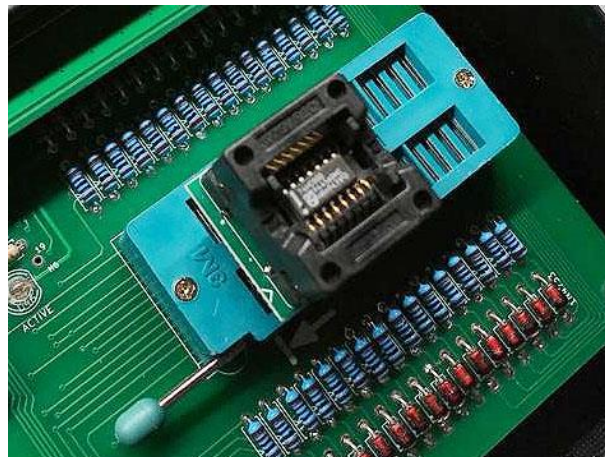


Abbildung 3.113: SOIC-16 Adapter

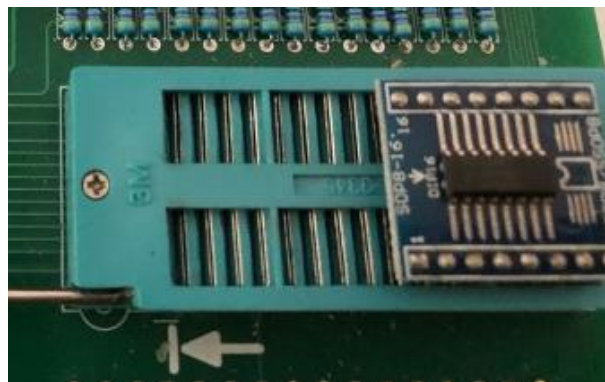


Abbildung 3.114: SOIC-16 Adapter

3.11 Einfacher Taktgenerator

Im Logik-Menü stellt der RCT einen einfachen Taktgenerator zur Verfügung. Der Takt wird an ZIF23 ausgegeben (5V, max. 8 mA) und umfasst den Frequenzbereich von 300 Hz bis 8 MHz.

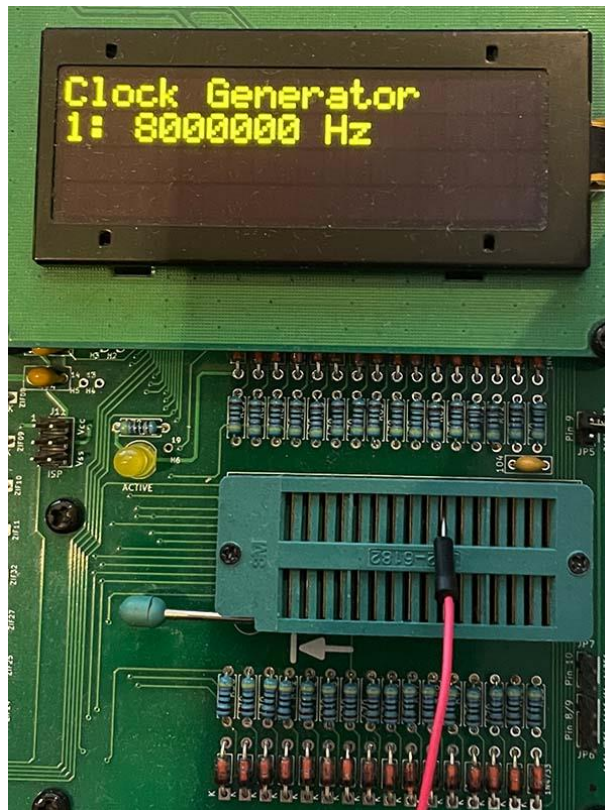


Abbildung 3.115: Taktgenerator

Der Takt wird vom Prozessor per PWM-Timer erzeugt, er wird vom 16 MHz Prozessortakt über interne Teiler abgeleitet, damit sind nur bestimmte Frequenzen möglich:

- 1.000, 1.333, 1.666, 2.000, 2.666, 3.200, 4.000, 5.333, 8.000 MHz
- 100, 166.666, 200, 266.666, 301.886, 320, 400, 500, 592.592, 666.666, 696.652, 800 kHz
- 10, 20, 30.018, 40, 50, 59.925, 69.868, 80, 89.887 kHz
- 1, 2, 3, 4, 5, 5.999, 6.999, 8, 8.998 kHz
- 300, 400, 500, 599, 700, 800, 899 Hz

Ein relativ gutes Rechtecksignal ist nur bis 1 MHz vorhanden.

Der Takt wird mit OK Ein- bzw. Ausgeschaltet.

3.11.1 Takterzeugung: nicht-überlappender Takt bzw. Quadratur-Takt

Mit Hilfe des Clock Generator Adapters (Taktgenerator Adapter) können aus dem bereitgestellten Taktsignal folgende zusätzliche Takte erzeugt werden:

- Takt und invertierter Takt, Pin C und /C
- nicht-überlappender Takt und invertierter Takt (non-overlapping clock), Pin C1 und C2
- Quadratur Takt (quadrature clock), Pin C_Q und C_E

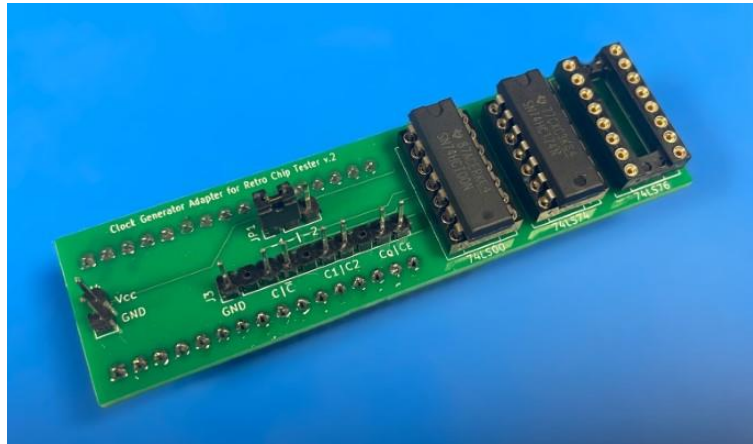


Abbildung 3.116: Taktgenerator Adapter

Es können Taktsignale bis zu 8 Mhz erzeugt werden. Der Quadraturtakt kann bis zu 2 MHz betragen.

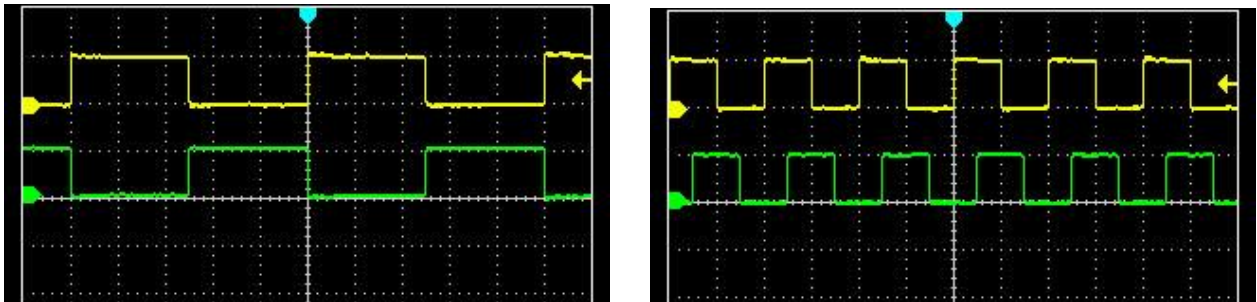


Abbildung 3.117: nicht-überlappend / invertierter Takt (links), Quadratur-Takt (rechts)

Der Jumper JP1 muss auf Pin 1-2 gesetzt werden oder sollte durch eine Drahtbrücke zwischen Pin 1-2 ersetzt werden.

An J4 muss die Spannungsversorgung (5V und GND) angelegt werden. Sie wurde absichtlich nicht schaltbar gestaltet, so dass keine Spannungsabfälle entstehen können.

3.12 Einfacher Frequenzzähler

Im Logikmenü stellt das RCT einen einfachen Frequenzzähler zur Verfügung, der mit und ohne speziellen Adapter verwendet werden kann.

Ohne Adapter:

Im Menü „Fin/2MHz“ wählen. Das Signal muss an ZIF25 anliegen (min. 2,5V, max. 5V) und deckt den Frequenzbereich von 1 Hz bis 2 MHz ab. Masse kann am Spannungssockel abgegriffen werden.

Mit „Frequency Counter“ Adapter:

Es gibt zwei Modi: „Fin“ und „Fdiv“. „Fin“ entspricht der Betriebsart ohne Adapter, „Fdiv“ nutzt einen Vorreiber, so dass höhere Frequenzen gezählt werden können (theoretisch bis 32 MHz). Der Adapter wird an den Spannungssockel zwischen Display und DC/DC-Modul (GND und 5V) angeschlossen.

Wenn Frequenzen bis zu 2 MHz gezählt werden sollen, den Jumper auf „Fin“ stellen und das Signal mit „Fin“ und „GND“ verbinden. Im Menü muss „Fin/2 MHz“ angezeigt werden.



Abbildung 3.118: Frequenzzähler („Fin“ Modus, bis 2 MHz)

Wenn Frequenzen bis 32 MHz gezählt werden sollen, den Jumper auf „Fdiv“ stellen und das Signal mit „Fdiv“ und „GND“ verbinden. Im Menü muss „Fdiv/xx MHz“ angezeigt werden (xx = 4, 8, 16, 32). Mit SELECT kann der gewünschte Modus ausgewählt werden. Mithilfe des Adapters wird das Signal zudem gefiltert und verstärkt.



Abbildung 3.119: Frequenzzähler („Fdiv“ Modus, bis 32 MHz)

Genauigkeit am Eingang „Fin/2MHz“:

Frequenz	Abweichung (Hz)
1 Hz	0 Hz
10 Hz	0 Hz
100 Hz	0 Hz
100 Hz – 100 kHz	0 Hz
100 kHz – 750 kHz	+/- 1 Hz
750 KHz – 1 MHz	+/- 2 Hz
1 MHz – 4 MHz	+/- 15 Hz
4 MHz – 7,5 MHz	+/- 30 Hz

Hinweise:

Der RCT kann am Eingang „Fin/2MHz“ Takte bis zu 7,5 MHz relativ genau zählen. Bei Frequenzen ab 2 MHz, steigt aber der Fehlerfaktor und das Taktsignal muss relativ sauber vorliegen. Mit der Adapterplatine als Vorteiler wird der Fehlerfaktor entsprechend höher.

Für die in der Tabelle genannten Werte, muss der Takt des ATmega2560 bei möglichst exakt 16 MHz liegen. Hierzu muss ein guter Quarz verwendet werden und ggf. die 22pF Kondensatoren angepasst werden (30 pF Trimmer verwenden).

4 Programmieren von EPROMs

Ab der Firmwareversion v.20 kann der RCT auch EPROMs programmieren. Zur Programmierung wird jeweils ein einfacher Adapter benötigt, der die notwendige Programmiervoltage zur Verfügung stellt. Weiterhin ist der SD-Kartenadapter notwendig und eine SD-Karte, welche die zu programmierende Binärdatei enthält.

Bitte beachten:

Der RCT ist in erster Linie ein Tester, kein Programmierer. Es ist aber nicht auszuschließen, dass in Zukunft weitere Bausteine unterstützt werden. Es wird sich dann aber hauptsächlich um ICs handeln, die von üblichen Programmierern nicht mehr unterstützt werden.



Es ist nicht garantiert, dass sich ein EPROM eines jeden Herstellers programmieren lässt. Der RCT unterstützt nur Standard-Algorithmen.

Die Stromaufnahme des EPROMs sollte nicht zu hoch sein. Idd (12V) sollte 50mA nicht überschreiten und die Verlustleistung sollte 800mW für längere Zeit nicht überschreiten.

Der RCT sollte beim Programmieren von EPROMs am besten über ein USB-Netzteil (mindestens 2A) versorgt werden. Eine Versorgung über den Hohlstecker ist zwar möglich, dabei kann aber - abhängig vom EPROM - der Linearregler (7805) auf der DC/DC-Platine überlastet werden (Temperatur beachten(!)). In diesem Fall regelt dieser ab und es kann zu einem Neustart des Testers kommen.

Vor dem Programmieren eines EPROMs muss die jeweilige Programmiervoltage eingestellt werden. Im nachfolgenden Kapitel wird dieser Vorgang beschrieben.

Bei der Handhabung mit den Programmieradaptern bitte darauf achten, keinen Kurzschluss mit anderen Bauteilen zu verursachen. Insbesondere beim Anschluss der Spannungsversorgung bitte äußerst sorgfältig vorgehen. Ein Kurzschluss bzw. Überspannung auf dem Mainboard kann den ATmega2560 zerstören.

Leertest:

Vor dem Programmieren wird ein Leertest des EPROMs durchgeführt. Dieser kann übersprungen werden, wenn die OK Taste lange gedrückt wird.

Kompatibilitätsmatrix:

Um ein EPROM zu programmieren, wird ein entsprechender Adapter benötigt, der die Programmiervspannung bereitstellt. Die folgende Tabelle zeigt, welche EPROMs programmiert werden können:

Programmierbare EPROMs			
EPROM	Menü	Benötigter Adapter	Spannung (Vpp)
2708	2708	2708 Adapter	26 V
2516	2716	2716/2532 Adapter	25 V
2758 [1]			25 V
2716			25 V
2716 B			12.5 V
HN48016 (*)			25 V
TMS2716	TMS2716	TMS2716 Adapter	25 - 27 V
2532	2532	2716/2532 Adapter	25 V
2732	2732	2732 Adapter	25 V
2732 A			21 V
2732 B			12.5 V
2564	2564	2564 Adapter	25 V

(*) EEPROM mit 25V Vpp

Bitte unbedingt das Datablatt bzgl. der erlaubten Spannung konsultieren.

Bemerkungen:

- [1] Der 2758 ist praktisch ein 2716, bei dem entweder die unteren oder oberen 1 kByte defekt sind und Ar (Pin 19) entsprechend gesetzt werden muss. Er lässt sich als 2716 mit einem Trick programmieren: Die 1 kByte müssen zweimal hintereinander in der Datei stehen. Damit wird der Chip dann einmal mit Ar=0 (A10=0) programmiert und einmal mit Ar=1 (A10=1). Ebenso sollte sich der TMS2508 so programmieren lassen. Hier wird dann aber die Speichermatrix gleich zweimal programmiert (Pin 19 ist nc). Dieses wurde aber noch nicht getestet.

4.1 Einstellen der Programmiervspannung (Vpp)

Beim Programmieren von EPROMs unbedingt auf die korrekte Programmiervspannung (Vpp) achten. Diese kann dem Datenblatt entnommen werden. Der Step-Up Regler des Programmieradapters muss möglichst genau auf diese Spannung eingestellt werden. Ist diese zu niedrig, schlägt die Programmierung fehl, ist diese zu hoch, kann der Speicher beschädigt werden (die maximalen Werte können dem Datenblatt entnommen werden).

Zum Einstellen von Vpp den Step-Up Regler zunächst mit 5V verbinden. Hierzu bieten sich zwei Varianten an:

1. Anschluss an eine externe 5V Spannungsquelle (der Adapter darf sich nicht im Tester befinden)
→ Die 5V werden für am Modul vorne links angeschlossen, GND/Vss vorne rechts

oder

2. Anschluss an die Sockelleiste (zwischen DC/DC-Modul und Display) an Vcc (5V) und GND.
→ Die Adapterplatinen verfügen über einen Vcc/GND Anschluss, der zum Anschluss verwendet werden sollte.

An den hinteren Anschlüssen des Step-Up Reglers muss mit einem Voltmeter die Programmiervspannung gemessen werden. Hierzu den (blauen) Trimmer entgegen des Uhrzeigersinns solange drehen, bis die passende Spannung angezeigt wird (die Spannung wird sich anfangs nicht ändern, sondern sich erst am Ende sprunghaft erhöhen).

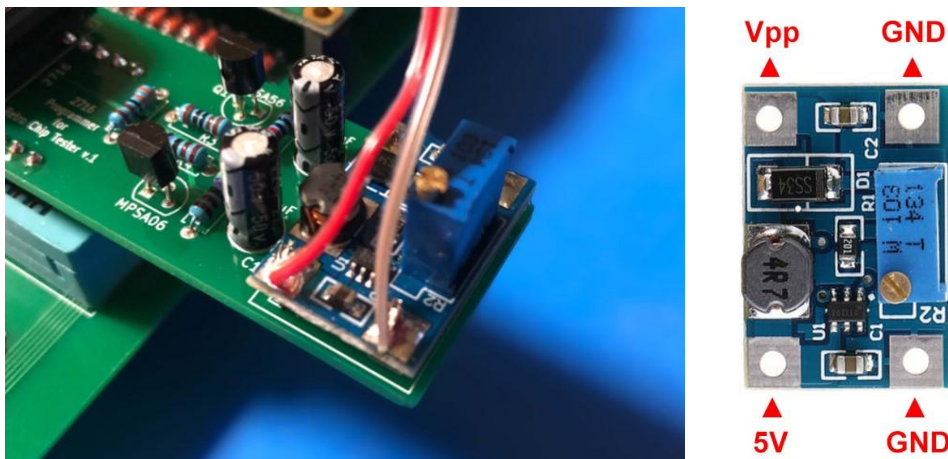


Abbildung 4.1: Anschluss der Eingangsspannung (5V)

Während der Programmierung kommt es je nach verwendetem Netzteil und Höhe der Stromaufnahme des EPROMs zu einem Spannungsabfall. Ist dieser zu hoch, schlägt die Programmierung fehl. In diesem Fall sollte die Programmiervspannung schrittweise leicht erhöht werden (ca. 1-2V). Es ist auch möglich die Programmiervspannung während eines Programmiervorgangs zu messen und entsprechend zu erhöhen, hierbei aber unbedingt darauf achten keinen Kurzschluss zu verursachen.

4.2 Auswahl der zu programmierenden ROM-Datei (ab FW v.23)

Nach Auswahl des EPROMs kann die zu programmierende ROM-Datei ausgewählt werden. Da sich möglicherweise mehrere Dateien auf der SD-Karte befinden, werden nur Dateien angezeigt, die den folgenden Kriterien entsprechen:

Dateien beginnend mit

- "2708." (wenn 2708 ausgewählt wurde)
- "2716." (wenn 2716 oder TMS2716 ausgewählt wurde)
- "2532." (wenn 2532 ausgewählt wurde)
- "2732." (wenn 2732 ausgewählt wurde)
- "2564." (wenn 2564 ausgewählt wurde)

sowie alle Dateien mit der Endung „.rom“ und „.bin“.

Die maximale Länge eines Dateinamens einschließlich der Erweiterung beträgt 20 Zeichen. Dateien mit längeren Dateinamen werden ignoriert. Die Dateiauswahl kann verlassen werden mit „FUNC/TOP“ oder am Ende der Liste mit „OK“ (bzw. „FUNC/TOP“).

4.3 2708 EPROM

Vorbereitung:

Die Programmierspannung des Adapters muss für die meisten 2708 EPROMs auf 25V – 27V eingestellt werden. Empfohlen wird die Programmierung zunächst mit 25V zu versuchen. **Bitte unbedingt das Datablatt bzgl. der erlaubten Spannung konsultieren.**



Abbildung 4.2: Programmieradapter

Der Adapter wurde ohne zusätzliche Modifikation mit EPROMs getestet, die beim Programmieren folgende maximale Stromaufnahme besitzen: $I_{cc} < 10 \text{ mA}$, $I_{dd} < 65 \text{ mA}$, $I_{bb} < 45 \text{ mA}$, $I_{prgm} < 20 \text{ mA}$.



Sollte die Stromaufnahme von I_{cc} bzw. I_{prgm} darüber liegen und die Programmierung dadurch fehlschlagen, muss Vcc und GND (befinden sich auf dem Step-up-Modul unten links (Vcc) und rechts (GND)) mit „[Vcc]“ und „Vss/GND“ an der Sockelleiste zwischen Display und DC/DC-Modul verbunden werden. Hierzu am Einfachsten die Drähte direkt an das Step-up Modul löten.

Programmieren:

Die Binärdatei muss unter dem Namen „rom2708.bin“ auf der SD-Karte abgelegt werden und genau 1024 Bytes enthalten. Ab FW v.23 kann diese Datei ausgewählt werden.

Anschließend wird im Untermenü **PROGRAMMING** der Baustein **2708** ausgewählt. Der Programmiervorgang startet sofort. Läuft alles fehlerfrei erscheint nach einiger Zeit „Finished“.



Abbildung 4.3: 2708 EPROM mit Adapter

Es können folgende Fehler auftreten:

- „rom2708.bin missing“: Die Binärdatei konnte nicht gefunden werden.
- „End of file error!“: Die Binärdatei enthält weniger als 1024 Bytes.
- „Verification failed“: Der Inhalt vom Speicherbaustein weicht vom Inhalt der Binärdatei ab.
- „Blank check failed“: Das zu programierende EPROM ist nicht leer.

Wenn die Überprüfung fehlschlägt, kann probiert werden den Baustein noch einmal zu programmieren. Der Leertest kann mit einem langen Druck auf OK übersprungen werden.

Die verwendeten Programmierparameter: $N = 256$, $t_{PW} = 390\mu\text{sec}$, $t_{SD} > 10\mu\text{sec}$, $t_{DH} > 1\mu\text{sec}$

4.4 2716/2516 EPROM and Hitachi HN48016 EEPROM

Vorbereitung:

Die Programmiervoltage des 2716/2532 Adapters muss für die meisten 2716/2516 EPROMs auf 21V oder 25V eingestellt werden. **Bitte unbedingt das Datablatt bzgl. der erlaubten Spannung konsultieren.**

Es ist unbedingt darauf zu achten, dass der Adapter zusätzlich mit „[Vcc]“ und „Vss/GND“ an der Sockelleiste zwischen Display und DC/DC Modul verbunden werden muss, da die EPROMs eine zu hohe Stromaufnahme besitzen.



Programmieren:

Die Binärdatei muss unter dem Namen „rom2716.bin“ auf der SD-Karte abgelegt werden und genau 2048 Bytes enthalten. Ab FW v.23 kann diese Datei ausgewählt werden.

Anschließend wird im Untermenü **PROGRAMMING** der Baustein **2716/2516** ausgewählt. Der Programmiervorgang startet sofort. Läuft alles fehlerfrei erscheint nach einiger Zeit „Finished“.



Abbildung 4.4: 2716 EPROM mit Adapter

Es können folgende Fehler auftreten:

- „rom2716.bin missing“: Die Binärdatei konnte nicht gefunden werden.
- „End of file error!“: Die Binärdatei enthält weniger als 2048 Bytes.
- „Verification failed“: Der Inhalt vom Speicherbaustein weicht vom Inhalt der Binärdatei ab.
- „Blank check failed“: Das zu programmierende EPROM ist nicht leer.

Wenn die Überprüfung fehlschlägt, kann probiert werden den Baustein noch einmal zu programmieren. Der Leertest kann mit einem langen Druck auf OK übersprungen werden.

Die verwendeten Programmierparameter: $N = 1$, $t_{PW} = 50\text{msec}$, $t_{SD} > 10\text{usec}$, $t_{DH} > 1\text{usec}$

Hinweise:

Ein „TMS2516“ ist **kompatibel** zum üblichen „2716“. Ein „TMS2716“ ist **nicht kompatibel** zum „2716“, hier werden mehrere Versorgungsspannungen benötigt.



Löschen des Hitachi **HN48016 EEPROMs** auf einem Steckbrett: 5V auf Pin 24, GND auf Pin 12 und 20 (/CS), 25V auf Pin 21. Ziehe Vpp (Pin 18) auf 25V (berühre Pin 18 und 21) für eine Sekunde. Das löscht den Chip.

Die Überprüfung des Inhalts erfolgt mit angelegter Vpp („Program Verify“). In Einzelfällen kann es bei älteren ICs vorkommen, dass diese Überprüfung (Verify) fehlschlägt. In diesem Fall kann das EPROM dennoch korrekt programmiert worden sein. Zur Kontrolle sollte das EPROM (ohne Adapter(!)) mit der normalen Auslesefunktion ausgelesen werden und dann der Inhalt verglichen werden.

4.5 TMS2716 EPROM (ab FW v.22)

Vorbereitung:

Die Programmierspannung des TMS2716 Adapters muss für die meisten TMS2716 EPROMs auf 25V – 27V eingestellt werden. **Bitte unbedingt das Datablatt bzgl. der erlaubten Spannung konsultieren.**

Es ist unbedingt darauf zu achten, dass der Adapter zusätzlich mit „[Vcc]“ und „Vss/GND“ an der Sockelleiste zwischen Display und DC/DC Modul verbunden werden muss, da die EPROMs eine zu hohe Stromaufnahme besitzen.



Programmieren:

Die Binärdatei muss unter dem Namen „**rom2716.bin**“ auf der SD-Karte abgelegt werden und genau 2048 Bytes enthalten. Ab FW v.23 kann diese Datei ausgewählt werden.

Anschließend wird im Untermenü **PROGRAMMING** der Baustein **TMS2716** ausgewählt. Der Programmiervorgang startet sofort. Läuft alles fehlerfrei erscheint nach einiger Zeit „Finished“.

Es können folgende Fehler auftreten:

- „rom2716.bin missing“: Die Binärdatei konnte nicht gefunden werden.
- „End of file error!“: Die Binärdatei enthält weniger als 2048 Bytes.
- „Verification failed“: Der Inhalt vom Speicherbaustein weicht vom Inhalt der Binärdatei ab.
- „Blank check failed“: Das zu programmierende EPROM ist nicht leer.

Wenn die Überprüfung fehlschlägt, kann probiert werden den Baustein noch einmal zu programmieren. Der Leertest kann mit einem langen Druck auf OK übersprungen werden.

Die verwendeten Programmierparameter: $N = 256$, $t_{PW} = 390\mu\text{sec}$, $t_{SD} > 10\mu\text{sec}$, $t_{DH} > 1\mu\text{sec}$

Hinweis:

Ein „TMS2516“ ist **kompatibel** zum üblichen „2716“. Ein „TMS2716“ ist **nicht kompatibel** zum „2716“, hier werden mehrere Versorgungsspannungen benötigt.



4.6 2532 EPROM

Vorbereitung:

Die Programmierspannung des 2716/2532 Adapters muss für die meisten 2532 EPROMs auf 25V eingestellt werden. **Bitte unbedingt das Datablatt bzgl. der erlaubten Spannung konsultieren.**

Es ist unbedingt darauf zu achten, dass der Adapter zusätzlich mit „[Vcc]“ und „Vss/GND“ an der Sockelleiste zwischen Display und DC/DC Modul verbunden werden muss, da die EPROMs eine zu hohe Stromaufnahme besitzen (siehe Abbildung 4.4).



Programmieren:

Die Binärdatei muss unter dem Namen „**rom2532.bin**“ auf der SD-Karte abgelegt werden und genau 4096 Bytes enthalten. Ab FW v.23 kann diese Datei ausgewählt werden.

Anschließend wird im Untermenü **PROGRAMMING** der Baustein **2532** ausgewählt. Der Programmiervorgang startet sofort. Läuft alles fehlerfrei erscheint nach einiger Zeit „Finished“.

Es können folgende Fehler auftreten:

- „rom2532.bin missing“: Die Binärdatei konnte nicht gefunden werden.
- „End of file error!“: Die Binärdatei enthält weniger als 4096 Bytes.
- „Verification failed“: Der Inhalt vom Speicherbaustein weicht vom Inhalt der Binärdatei ab.
- „Blank check failed“: Das zu programmierende EPROM ist nicht leer.

Wenn die Überprüfung fehlschlägt, kann probiert werden den Baustein noch einmal zu programmieren. Der Leertest kann mit einem langen Druck auf OK übersprungen werden.

Die verwendeten Programmierparameter: $N = 1$, $t_{PW} = 50\text{msec}$, $t_{SD} > 10\text{usec}$, $t_{DH} > 1\text{usec}$

4.7 2732 EPROM

Vorbereitung:

Die Programmierspannung des 2732 Adapters muss für die meisten 2732 EPROMs auf 25V und für 2732A auf 21V eingestellt werden. **Bitte unbedingt das Datablatt bzgl. der erlaubten Spannung konsultieren.**

Es ist unbedingt darauf zu achten, dass der Adapter zusätzlich mit „[Vcc]“ und „Vss/GND“ an der Sockelleiste zwischen Display und DC/DC Modul verbunden werden muss, da die EPROMs eine zu hohe Stromaufnahme besitzen (siehe Abbildung 4.4).



Programmieren:

Die Binärdatei muss unter dem Namen „rom2732.bin“ auf der SD-Karte abgelegt werden und genau 4096 Bytes enthalten. Ab FW v.23 kann diese Datei ausgewählt werden.

Anschließend wird im Untermenü **PROGRAMMING** der Baustein **2732** ausgewählt. Der Programmiervorgang startet sofort. Läuft alles fehlerfrei erscheint nach einiger Zeit „Finished“.



Abbildung 4.5: 2732 EPROM mit Adapter

Es können folgende Fehler auftreten:

- „rom2732.bin missing“: Die Binärdatei konnte nicht gefunden werden.
- „End of file error!“: Die Binärdatei enthält weniger als 4096 Bytes.
- „Verification failed“: Der Inhalt vom Speicherbaustein weicht vom Inhalt der Binärdatei ab.
- „Blank check failed“: Das zu programmierende EPROM ist nicht leer.

Wenn die Überprüfung fehlschlägt, kann probiert werden den Baustein noch einmal zu programmieren. Der Leertest kann mit einem langen Druck auf OK übersprungen werden.

Die verwendeten Programmierparameter: $N = 1$, $t_{PW} = 50\text{msec}$, $t_{SD} > 10\text{usec}$, $t_{DH} > 1\text{usec}$

4.8 2564 EPROM (ab FW v.22)

Vorbereitung:

Die Programmierspannung des 2564 Adapters muss für die meisten 2564 EPROMs auf 25V eingestellt werden. **Bitte unbedingt das Datablatt bzgl. der erlaubten Spannung konsultieren.**

Es ist unbedingt darauf zu achten, dass der Adapter zusätzlich mit „[Vcc]“ und „Vss/GND“ an der Sockelleiste zwischen Display und DC/DC Modul verbunden werden muss, da die EPROMs eine zu hohe Stromaufnahme besitzen (siehe Abbildung 4.4).



Programmieren:

Die Binärdatei muss unter dem Namen „rom2564.bin“ auf der SD-Karte abgelegt werden und genau 8192 Bytes enthalten. Ab FW v.23 kann diese Datei ausgewählt werden.

Anschließend wird im Untermenü **PROGRAMMING** der Baustein **2564** ausgewählt. Der Programmiervorgang startet sofort. Läuft alles fehlerfrei erscheint nach einiger Zeit „Finished“.

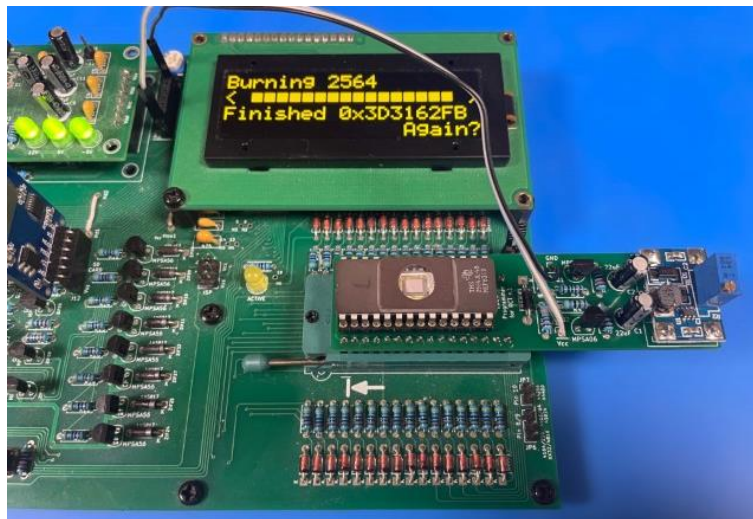


Abbildung 4.6: 2564 EPROM mit Adapter

Es können folgende Fehler auftreten:

- „rom2564.bin missing“: Die Binärdatei konnte nicht gefunden werden.
- „End of file error!“: Die Binärdatei enthält weniger als 8192 Bytes.
- „Verification failed“: Der Inhalt vom Speicherbaustein weicht vom Inhalt der Binärdatei ab.
- „Blank check failed“: Das zu programmierende EPROM ist nicht leer.

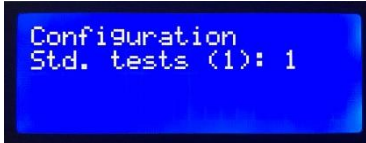
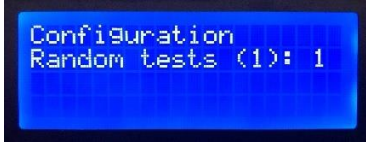
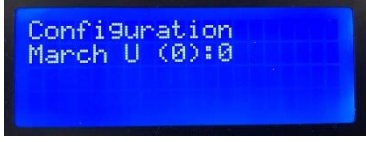
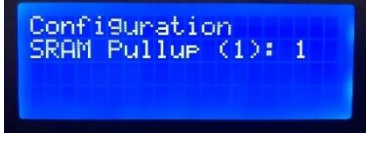
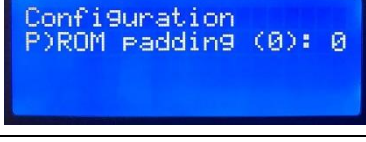
Wenn die Überprüfung fehlschlägt, kann probiert werden den Baustein noch einmal zu programmieren. Der Leertest kann mit einem langen Druck auf OK übersprungen werden.

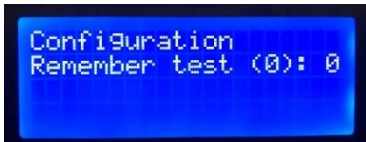
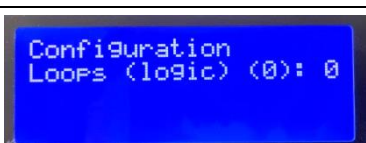
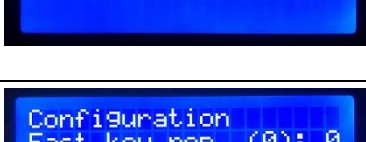
Die verwendeten Programmierparameter: $N = 1$, $t_{PW} = 50\text{msec}$, $t_{SD} > 10\text{usec}$, $t_{DH} > 1\text{usec}$

5 Einstellungs­menü / Konfigurations­menü

Über das Konfigurationsmenü lassen sich bestimmte Einstellungen vornehmen, die dauerhaft im EEPROM des ATmega2560 gespeichert werden.

Damit diese Einstellung auch nach einem Neuprogrammieren des ATmega2560 noch vorhanden sind, müssen die Fuses entsprechend gesetzt werden. Allerdings kann es auch sinnvoll sein, dass das EEPROM bei einem Update der Firmware ebenfalls gelöscht wird.

Aktivieren der SD-Karten Unterstützung. Mit aktivierter Unterstützung für SD-Karten können, mit Hilfe eines geeigneten SD-Kartenmoduls, ROMs und EPROMs auch ausgelesen werden. Standardeinstellung: AUS (0)	
Durchführen der Standard Tests (Pattern Tests, Counting Tests). Das sind die Standard Tests, die i.d.R. durchgeführt werden. Standardeinstellung: EIN (1)	
Durchführen von Tests mit Zufallszahlen für SRAMs und DRAMs. Diese Tests verlangsamen den Test eines Chips erheblich, deshalb können sie abgeschaltet werden. Standardeinstellung: EIN (1) - Nur in Kombination mit „Std. tests“.	
Durchführen von Tests nach dem „March Y“ Algorithmus. Anstelle (oder zusätzlich) zu den Standard Tests, kann dieser Test ausgeführt werden. Standardeinstellung: AUS (0)	
Durchführen von Tests nach dem „March U“ Algorithmus. Anstelle (oder zusätzlich) zu den Standard Tests, kann dieser Test ausgeführt werden. Standardeinstellung: AUS (0)	
Die Datenleitungen von SRAMs werden mit Pullups beschaltet. Standardeinstellung: EIN (1)	
Die Datenleitungen von DRAMs werden mit Pullups beschaltet. Standardeinstellung: AUS (0)	
Die Datenleitungen von (P)ROMs und EPROMs werden mit Pullups beschaltet. Standardeinstellung: EIN (1)	
Ist diese Funktion eingeschaltet, versucht der Tester vorab zu erkennen, ob ggf. Pull Ups für DRAMs oder SRAMs benötigt werden (nur, wenn Pullups für DRAMs bzw. SRAMs deaktiviert sind). [*] Standardeinstellung: AUS (0)	
(P)ROMs und EPROMs können mit ‚0‘ oder ‚1‘ gepaddet werden, sollte ein Wort weniger als 8 Bit besitzen. Standardeinstellung: ‚0‘-Padding (0)	

Beim Programmieren von EPROMs können Zellen mit 0xFF (leere Bereiche) übersprungen werden. Standardeinstellung: AUS (0)	
Ist diese Funktion aktiviert, wird versucht die DRAM Features (Page Mode, Static Column Mode) zu ermitteln. [*] Standardeinstellung: AUS (0)	
Der Speichertester merkt sich den zuletzt getesteten Chip. Standardeinstellung: AUS (0)	
Hiermit wird festgelegt, wie oft ein Logik-Test wiederholt wird, wenn dieses per langem Druck auf OK ausgelöst wird. Die Anzahl entspricht 2^n, wobei n der eingestellte Wert ist. Standardeinstellung: AUS (0), 1 = 2 Durchläufe, 2 = 4 Durchläufe, 3 = 8 Durchläufe, ... (ab 6 schaltet der Tester in einen schnellen Modus), 11 = unendlich viele Durchläufe	
Ein-/Ausschalten von Hinweistönen und Tastenklicks. Standardeinstellung: EIN (1)	
Festlegen eines alternativen Testmodus (siehe Tabelle unten). Standardeinstellung: AUS (0)	
Der Tester prüft vor Ausführung eines Tests, ob der korrekte IC eingesetzt wurde bzw. dieser testbar ist. (0) = Aus (1) = EIN (Standardeinstellung) (2) = zusätzlich testen, ob CE/CS korrekt funktioniert (nur bei SRAM)	
Die Geschwindigkeit der Tastenwiederholung wird erhöht. Standardeinstellung: AUS (0)	
Die selbstdefinierten SRAM und DRAM-Speicherbausteine werden gelöscht.	
Die Konfiguration auf die Standardwerte zurücksetzen.	
Beenden des Einstellungsmenüs und speichern der Einstellungen.	

[*] Diese Funktionen sind experimentell. Sollten Fehler in Zusammenhang mit diesen auftreten, sollte die Standardeinstellung verwendet werden.

Bedeutung des **alternativen Testmodus** (wird ausgeführt bei langem Tastendruck auf OK):

Wert	Bedeutung
0	Kein alternativer Modus festgelegt.
1	DUMP Modus: Ein langer Tastendruck auf OK führt einen erzwungenen Standard-Test für SRAMs bzw. DRAMs aus und speichert einen Dump der gelesenen Werte auf SD-Karte. Anzeige „DMP“ im Display. Dieser Modus muss nicht aktiviert werden, wenn ROMs/EPROMs gedumpt werden sollen! Er beeinflusst nur das Verhalten bzgl. SRAMs und DRAMs.
2	Nur für 4164 DRAM: Verwendet den RMW-Modus anstelle des Early-Write-Modus beim Schreiben von Daten. Anzeige „RMW“ im Display.

6 Warnhinweise

Warnhinweise zur Durchführung von Tests:

Die Chips immer **nach links** (im Sockel: „oben“) bündig einsetzen!

Die Spannungsversorgung wird erst während des Tests eingeschaltet.

Die Ports des ATmega2560 sind gegen Kurzschlüsse gesichert, d.h. der ATmega2560 sollte bei einem solchen Defekt keinen Schaden nehmen. I.d.R. passiert bei defekten Chips nichts, es besteht aber immer eine (sehr geringe) Gefahr.

Warnhinweise zur Spannungsversorgung des Speichertesters:

Der Tester sollte mit einem Netzteil betrieben werden. Zwar wird die Stromaufnahme selten über 500mA liegen, aber insbesondere die älteren Chips benötigen schon einmal 180-200mA bei 5V, was zusammen mit LCD, ATmega2560 und den LEDs schon einmal über 500mA liegen kann. Eine zu hohe Stromaufnahme kann einen USB-Port beschädigen. Sollten Kurzschlüsse auftreten, können auch diese einen USB-Port beschädigen.



Die Polyfuse sichert den USB-Port gegen Ströme >1100mA ab. Die Polyfuse kann notfalls entfallen und kurzgeschlossen werden. Wird die Polyfuse nicht verwendet, sollte der Tester nur mit einem Netzteil betrieben werden.

Warnhinweise zur Programmierung des Speichertesters:

Niemals den Speichertester **gleichzeitig** mit dem ISP-Programmierer und per USB, Hohlstecker oder Schraubterminal mit Spannung versorgen.

Wenn mit dem gesteckten DC/DC-Konverter „Alternative #2“ (mit Recom-Wandler) der ATmega2560 per ISP programmiert wird, ist zu beachten, dass der DC/DC-Konverter bei einer Spannungsversorgung durch den Programmierer, die Versorgungsspannungen Vdd und Vbb erzeugt. Der Programmierer sollte diese zusätzliche Last vertragen können, ansonsten sollte der DC/DC-Konverter vorher entfernt werden. Bei dem oben empfohlenen Programmierer ist das der Fall und es können sogar Chips getestet werden.

7 Bekannte Probleme und Einschränkungen

Vorab: Der Tester kann nicht alle Probleme erkennen, aber ziemlich viele. ;)

7.1 Erkennung von Fehlern (Speicherchips)

Der Retro Chip Tester wurde entwickelt, um „alte“ Speicherchips aus den 1970er und 1980er Jahren testen zu können, die von heutigen Programmiergeräten oft nicht mehr erkannt werden. Es wird häufig empfohlen Chips in einem gleichen Gerät einzusetzen und so die Funktionsfähigkeit zu testen. Dabei wird vergessen, dass die übrige Hardware ebenfalls schon entsprechend alt ist und ein häufiges Ein- und Ausschalten weitere Fehler provozieren kann.

Ein idealer Speichertester sollte natürlich

- defekte Speicherzellen,
- Timing-Fehler durch Materialermüdung (z.B. bei DRAMs) und
- fehlerhafte Signalpegel durch Materialermüdung

erkennen.

Leider ist ein solcher Tester nicht zu einem akzeptablen Preis herstellbar, ein realer Tester wird immer ein Kompromiss zwischen der Erkennungsquote von defekten Chips und dem Preis sein.

Defekte Speicherzellen:

Im Grunde sollten folgende Fehler erkannt werden können:

- Stuck-At Fault
Der logische Wert einer Zelle ist immer 0 oder 1.
- Transition Fault
Ein Wechsel von $0 \rightarrow 1$ oder $1 \rightarrow 0$ schlägt fehl.
- Coupling Fault
Eine Schreiboperation in eine Zelle ändert den Inhalt einer zweiten Zelle.
- Neighborhood Pattern Sensitive Fault
Der Inhalt einer Zelle wird durch den Inhalt einer anderen Zelle beeinflusst.
- Address Decoder Fault
Jeder Fehler, der mit dem Adressdecoder im Zusammenhang steht (kein Zugriff möglich, Ändern mehrerer Zellen gleichzeitig, Zugriff auf eine Zelle von mehreren Adressen aus)

[Lindner13]⁷ unterscheidet diese Fehler (bezogen auf SRAMs) detaillierter in Fehler von einer Zelle (Single-cell fault primitives) und von zwei Zellen (two-cell fault primitives), also verbundenen Zellen. In den folgenden zwei Absätzen wird dessen ausführlichere Beschreibung zitiert.

Fehler einzelner Zellen sind:

- State Faults: Der Wert der Zelle wird ohne Leseoperation verändert und hängt vom Anfangszustand der Zelle ab.
- Transition Faults: Die Zelle kann nicht geändert werden, wenn sie mit dem entgegengesetzten Wert geschrieben wird. Das heißt, Übergang $0 \rightarrow 1$ oder $1 \rightarrow 0$.
- Write Destructive Faults: Eine Schreiboperation ohne Übergang ($0 \rightarrow 0$ oder $1 \rightarrow 1$) verursacht einen Übergang.
- Read Destructive Faults: Eine Leseoperation bewirkt, dass die Zelle umgedreht wird und der falsche Wert zurückgegeben wird.

⁷ [Lindner13] Test Set Optimization for Industrial SRAM Testing, Michael Linder, Technische Universität München, 2013, <http://d-nb.info/1045729981>

- Deceptive Read Destructive Faults: Eine Leseoperation bewirkt, dass die Zelle ihren Wert ändert, jedoch die korrekte Ausgabe zurückgegeben wird.
- Random Read Destructive Faults: Eine Leseoperation dreht die Zelle um und ein zufälliger Logikwert wird an die Ausgabe zurückgegeben.
- Incorrect Read Faults: Eine Leseoperation gibt den falschen Wert zurück. Der in der Zelle gespeicherte Wert bleibt jedoch korrekt.
- Random Read Faults: Eine Leseoperation gibt einen zufälligen Logikwert zurück, während der gespeicherte Wert korrekt bleibt.
- Undefined State Faults: Ohne Leseoperation wechselt der Logikwert einer Zelle in einen undefinierten Zustand.
- Undefined Write Faults: Ein undefinierter Zustand der Zelle wird durch eine Schreiboperation verursacht.
- Undefined Read Faults: Die Zelle wird durch eine Leseoperation in einen undefinierten Zustand versetzt.
- Stuck-At Faults: Die Zelle bleibt für jede Operation auf einem festen Wert.
- No Access Faults: Auf die Zelle kann nicht zugegriffen werden. Eine Schreiboperation kann den Wert der Zelle nicht ändern und eine Leseoperation gibt einen zufälligen Wert zurück. Dieser Fehler muss nicht durch den Adressdecoder verursacht werden, sondern kann auch durch eine offene Wortleitung verursacht werden.
- Data Retention Faults: Der Wert einer Zelle ändert sich nach einer bestimmten Zeit, ohne auf die Zelle zuzugreifen.

Fehler verbundener Zellen sind:

- State Coupling Faults: Die v-Zelle wird in einen gegebenen logischen Zustand gezwungen, wenn sich die a-Zelle in einem gegebenen logischen Zustand befindet, ohne eine Operation an der v-Zelle oder der a-Zelle auszuführen.
- Undefined State Coupling Faults: Der Zustand der v-Zelle ist undefiniert, während sich die a-Zelle in einem bestimmten logischen Zustand befindet, ohne dass eine Operation an der v-Zelle oder der a-Zelle ausgeführt wird.
- Disturb Coupling Faults: Jede Operation, die an der a-Zelle ausgeführt wird, bewirkt, dass die v-Zelle kippt.
- Undefined Disturb Coupling Faults: Jede an der a-Zelle ausgeführte Operation zwingt die v-Zelle in einen undefinierten Zustand.
- Idempotent Coupling Faults: Eine Übergangsschreiboperation für die a-Zelle bewirkt, dass die v-Zelle umgedreht wird.
- Inversion Coupling Faults: Eine Übergangsschreiboperation für die a-Zelle invertiert den logischen Wert der v-Zelle.
- Transition Coupling Faults: Ein gegebener Logikwert in der a-Zelle verursacht eine fehlgeschlagene Übergangsschreiboperation, die an der v-Zelle ausgeführt wird.
- Write Destructive Coupling Faults: Ein gegebener logischer Zustand der a-Zelle verursacht einen Übergang in der v-Zelle, obwohl eine nicht übergangsbedingte Schreiboperation an der v-Zelle ausgeführt wird.
- Read Destructive Coupling Faults: Wenn sich die a-Zelle in einem bestimmten Zustand befindet, ändert eine Leseoperation an der v-Zelle ihren Wert und gibt den falschen Wert an den Ausgang zurück.
- Deceptive Read Destructive Coupling Faults: Wenn sich die a-Zelle in einem bestimmten Zustand befindet, ändert eine Leseoperation an der v-Zelle ihren Wert und der korrekte Wert wird an den Ausgang zurückgegeben.
- Random Read Destructive Coupling Faults: Wenn sich die a-Zelle in einem bestimmten Zustand befindet, ändert eine Leseoperation an der v-Zelle den Wert in der v-Zelle und ein zufälliger Wert wird an den Ausgang zurückgegeben.

- **Incorrect Read Coupling Faults:** Wenn sich die a-Zelle in einem bestimmten Zustand befindet, gibt eine Leseoperation an der v-Zelle den falschen Wert an den Ausgang zurück.
- **Random Read Coupling Faults:** Wenn sich die a-Zelle in einem bestimmten Zustand befindet, gibt eine Leseoperation an der v-Zelle einen zufälligen Wert an den Ausgang zurück, während der Wert der v-Zelle korrekt bleibt.
- **Undefined Write Coupling Faults:** Eine Schreiboperation für die v-Zelle zwingt sie in einen undefinierten Zustand, während sich die a-Zelle in einem bestimmten Zustand befindet.
- **Undefined Write Coupling Faults:** Eine Leseoperation an der v-Zelle zwingt sie in einen undefinierten Zustand, während sich die a-Zelle in einem bestimmten Zustand befindet. Der an die Ausgabe zurückgegebene Wert kann korrekt, falsch oder zufällig sein.

Der Tester verwendet standardmäßig traditionelle Tests: Die Chips werden vom Tester mit verschiedenen Patterns beschrieben, sollte eine Speicherzelle defekt sein, wird sie hierdurch i.d.R. entdeckt. Weiterhin werden die Zellen mit einem auf- und absteigenden Muster beschrieben, so kann entdeckt werden, ob ggf. eine Adressleitung ausgefallen ist. Der langsamste (aber von diesen auch zuverlässigste) Test ist der Test mit Zufallszahlen. Wer Chips schnell testen möchte, kann diesen Test in der Konfiguration abschalten.

Diese traditionellen Tests erkennen die o.g. Fehler i.d.R. ziemlich gut. Zur Geschwindigkeitsoptimierung von Speichertests wurden mehrere Algorithmen entwickelt, wie z.B. die „March“-Tests. Der Chip-Tester unterstützt aktuell den „March Y“- und „March U“-Test.

Der „March-Y“ Test ist schneller als die traditionellen Tests zusammen. Er ist ein guter Kompromiss aus Erkennungsgenauigkeit und Geschwindigkeit. Wer möchte kann diesen Test-Algorithmus anstelle oder zusätzlich zu den traditionellen Tests in der Konfiguration aktivieren. Der Test verläuft nach folgender Sequenz: { $\uparrow(w0)$; $\uparrow(r0,w1,r1)$; $\downarrow(r1,w0,r0)$; $\downarrow(r0)$ }. Die Laufzeit beträgt 8n.

Der „March-U“ Test ist langsamer als die traditionellen Tests zusammen, aber erkennt Fehler sehr zuverlässig. Wer möchte kann diesen Test-Algorithmus anstelle oder zusätzlich zu den traditionellen Tests in der Konfiguration aktivieren. Der Test verläuft nach folgender Sequenz: { $\uparrow(w0)$; $\uparrow(r0,w1,r1,w0)$; $\uparrow(r0,w1)$; $\downarrow(r1,w0,r0,w1)$; $\downarrow(r1,w0)$; $\downarrow(r0)$ }. Die Laufzeit beträgt 14n.

Der Test auf defekte Speicherzellen ist sehr zuverlässig und identifiziert i.d.R. die meisten defekten RAMs.

Die folgende Tabelle gibt eine Übersicht, welche Fehler von welchem Test identifiziert werden können.

	Pattern	Alt/Count	Rnd	March-Y	March-U
Stuck-At Fault	X	(X)	(X)	X	X
Transition Fault	X	(X)	(X)	X	X
Coupling Fault		(X)	X	X	X
Neighborhood Pattern Sensitive Fault		(X)	X	X	X
Address Decoder Fault		(X)	X	X	X

Timing-Fehler:

Genaue Timing-Tests können mit einem 16MHz getakteten ATmega2560 nicht durchgeführt werden. Hier wäre eine Auflösung von mindestens ca. 1-5ns notwendig, die mit einem ATmega2560, der für einen Befehl schon 62,5ns benötigt, nicht realisierbar sind. Somit können Taktflanken und die Einhaltung von Timings nicht gemessen werden. Auch der Refresh bei DRAMs wird nicht explizit getestet, sondern nur implizit indem aufeinanderfolgend auf Speicherzellen zugegriffen wird (der Zyklus dauert länger als 2ms). I.d.R. lässt dieses schon die Aussage zu, ob eine Speicherzelle überhaupt noch in der Lage ist Daten zu speichern. Timing-Fehler sind aber recht selten, meistens sind die Speicherzellen selbst defekt.

Der Tester spricht alle Chips mit einem recht konservativen (=langsamen) Timing an. Sollte ein Chip mit diesen Timings Probleme haben, ist es sehr wahrscheinlich, dass er defekt ist. Der Fehler selbst ist nicht von einem Bitfehler zu unterscheiden und wird vom Tester entsprechend ausgegeben.

Fehlerhafte Signalpegel:

Es kann durchaus sein, dass ein Chip, der in einem Computer (so gerade eben) noch korrekt läuft, vom Tester als fehlerhaft erkannt wird. Wird der Tester per USB betrieben, liegt die Versorgungsspannung i.d.R. bei 4,7 bis 4,8V. Es gibt Chips, die, wenn sie mit weniger als 5V betrieben werden, aufgrund der Alterung nicht mehr fehlerfrei arbeiten. Das ist dann nicht mehr i.O., denn i.d.R. sind +/- 10% bei Vcc zulässig.

Ein weiteres Problem, welches mit einem ATmega2560 nicht zufriedenstellend gelöst werden kann, ist die Bestimmung der korrekten Spannungspegel. Da die Chips nur digital angesprochen werden, kann nicht ermittelt werden, ob ein Chip die Spannungspegel korrekt einhält. Theoretisch könnte man die Analogports dafür verwenden, aber dieses wäre aufgrund der 16 Ports nur bei wenigen Chips einsetzbar und die Ports müssten aufwendig kalibriert werden, was die Hardwarekosten steigen lässt.

Ein weiteres Problem sind die unterschiedlichen Spannungspegel vom ATmega2560 und dem zu testenden IC: Laut Datenblatt sollte ein 4164 für ein LOW max. 0.8V als Pegel liefern, für ein HIGH min. 2.4V. Der ATmega2560 erkennt bei 5V Versorgungsspannung Pegel unter 1.5 V als LOW, Pegel über 3 V als HIGH. Chips, die also knapp über 2.4V an einem Ausgang liefern, werden vom Tester ggf. als fehlerhaft angezeigt, obwohl sie noch innerhalb der Parameter liegen (allerdings so knapp, dass der Chip nicht mehr verwendet werden sollte).

Weitere Tests:

Grundsätzlich wären weitere spezielle Tests sinnvoll:

- Kurzschlüsse zwischen Signalleitungen oder unterbrochene Signalleitungen.
- Stromverbrauchstest
- Test von Ausgangsströmen

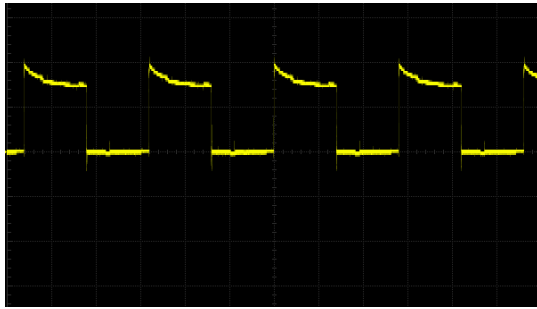
Der Chip-Tester erkennt Kurzschlüsse zwischen Signalleitungen oder unterbrochene Signalleitungen durch die traditionellen Tests als Bitfehler.

Ein Stromverbrauchstest kann durch die verwendete Hardware nicht durchgeführt werden. Bei diesem Test wird davon ausgegangen, dass defekte Chips u.U. einen höheren Stromverbrauch haben.

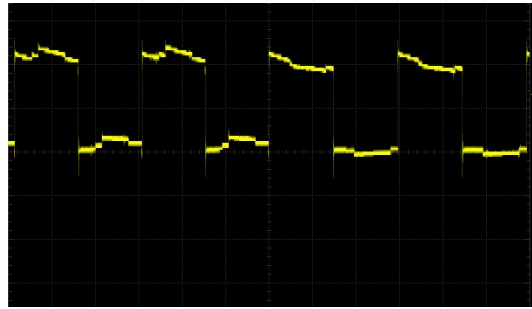
Ein Testen der möglichen Ausgangsströme kann durch die verwendete Hardware nicht durchgeführt werden. Bei diesem Test könnte festgestellt werden, ob Ausgänge noch genug Strom zum Schalten von Eingängen liefern.

Wie unterschiedlich die Signale am Ausgang z.B. eines 4164 sein können, zeigen die folgenden Bilder:

Alternierend 0 und 1 gelesen:

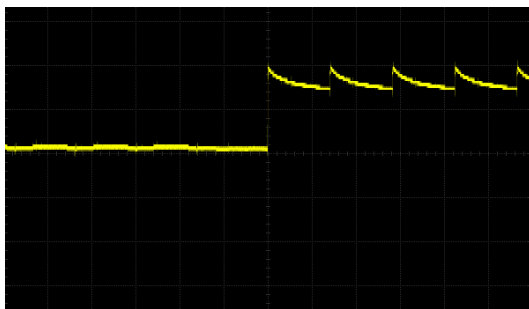


IC 1: D_{out}, Pegel zwischen 3V und 4V

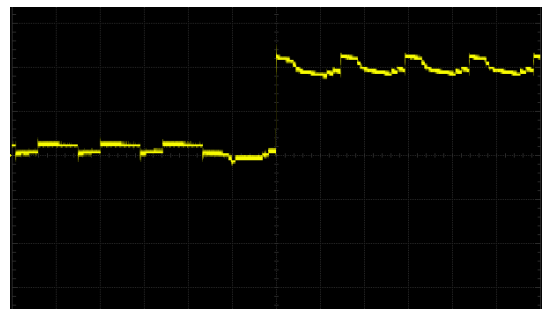


IC 2: D_{out}, Pegel zwischen 4V und 4.5V

Wechsel von „nur 0“ auf „nur 1“:



IC 1: D_{out}, Pegel zwischen 3V und 4V



IC 2: D_{out}, Pegel zwischen 3,9V und 4.6V

Die Spannungspegel eines HIGH liegt bei beidem ICs im erlaubten Bereich. IC 2 liefert beim Lesen eines LOW ein Wert bis zu 0,5V. Das liegt oberhalb von dem, was erlaubt ist. Da die Pegel bei IC 2 sehr variabel sind, können Tests teilweise als fehlerfrei durchlaufen, i.d.R. sollte der Chip aber als fehlerhaft erkannt werden.

In einem realen System (hier am Beispiel eines Oric-1 Heimcomputers) sind die Signalpegel sehr viel schlechter (wobei die im Bild gezeigten Pegel im Vergleich sehr gut sind). Im Bild werden die Signale für RAS (Violett), CAS (Blau) und WE (Gelb) dargestellt (4164 DRAM). Sehr schön zu erkennen ist der Speicherzyklus: RAS wird aktiv (low), 75ns später folgt CAS. RAS bleibt 175ns aktiv, CAS bleibt 250ns aktiv. CAS wird nach RAS, kurz bevor der nächste Speicherzyklus beginnt, inaktiv.

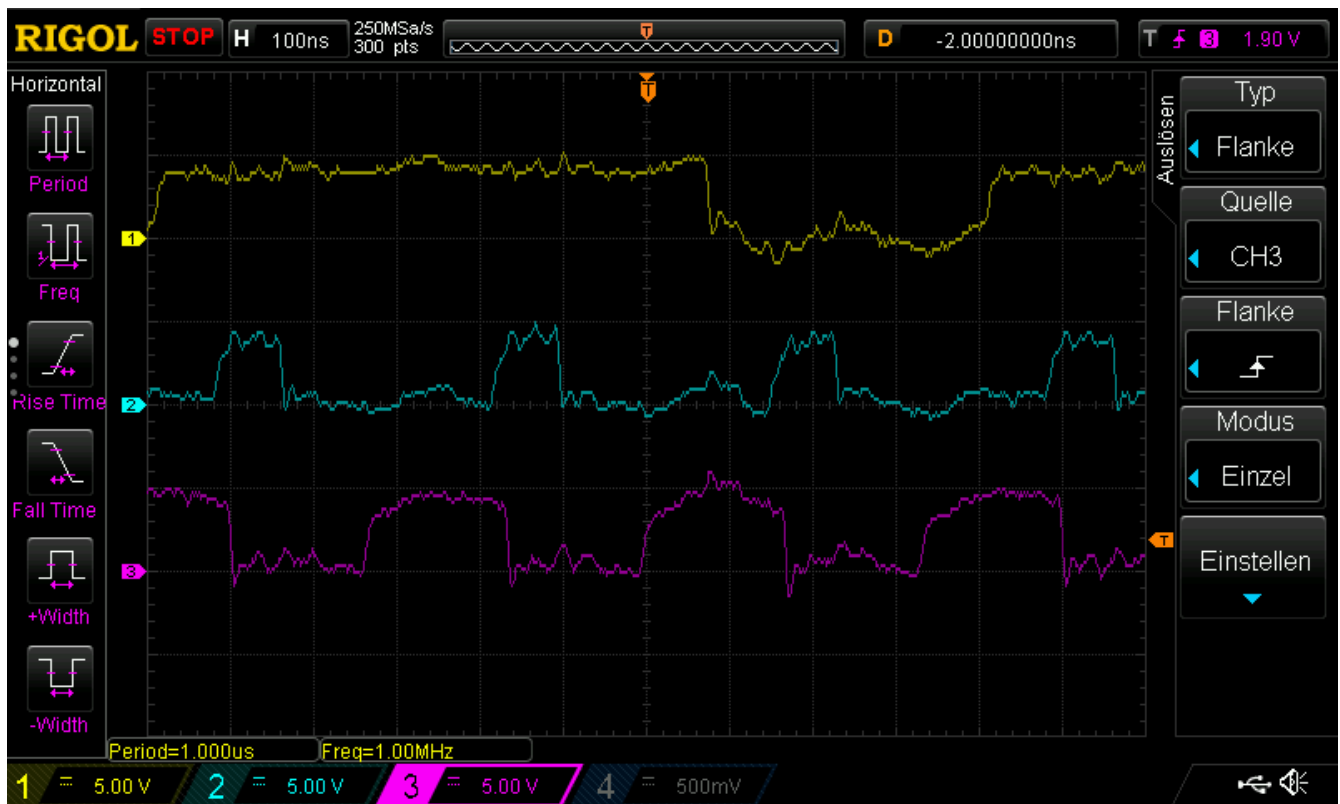


Abbildung 7.1: RAS/CAS/WE Timing (Oric-1)

Ein IC der unter optimalen Bedingungen im Tester als fehlerfrei getestet wird, könnte in einem realen System aufgrund der Störungen durchaus Fehler aufweisen.

Allgemeine Aussagekraft von Speichertests

Es stellt sich auch immer die Frage, ob bestimmte Tests für den gedachten Einsatzzweck sinnvoll sind.

Ein Beispiel:

Nehmen wir einen 120ns SRAM-Chip an, wie er in einem ZX Spectrum oder C64 verwendet wird. Nach 40 Jahren sind die Speicherzellen (hoffentlich) immer noch zu 100% in Ordnung, aber die Zugriffszeit beträgt jetzt 150ns. Was soll ein Tester als Ergebnis ausgeben?

- a) „FAIL“, weil der Chip die 120ns aus der Spezifikation nicht mehr erfüllt? In einem ZX Spectrum oder C64 funktioniert dieser aber immer noch sehr gut und ohne Tester wäre es vermutlich nie aufgefallen, dass dieser Chip „defekt“ ist. Ja genau: "defekt", da die Spezifikation des Chips nicht eingehalten wurde. Da der Tester ein FAIL ausgegeben hat, wird der Chip weggeworfen?

oder

- b) „PASS“, weil die Speicherzellen noch alle in Ordnung sind und der Chip in 99,99% aller Fälle auch ohne korrekte Timings weiterhin funktioniert? Wenn bei defekten Speicherzellen ein FAIL ausgegeben wird, ist er definitiv defekt und kann auch entsorgt werden.

Sie können sich nicht nur auf ein Testergebnis, wie FAIL oder PASS verlassen. Es hängt von der Anwendung ab. Bei einem sicherheitskritischen System wird hingegen vermutlich gefordert werden, dass ein Chip seine technische Spezifikation exakt einhält.

Wird ein IC vom Tester mit „PASS“ bewertet, so wird dieser mit sehr hoher Wahrscheinlichkeit in Ordnung sein, umgekehrt kann ein mit „FAIL“ bewerteter IC noch funktionieren, wird es aber vermutlich nicht oder zumindest nicht auf Dauer. Diese „Grenzfälle“ sind aber relativ selten.

Selbst zwei gleich aufgebaute Tester können beim selben IC, wenn es sich um einen „Grenzfall“ handelt, zu unterschiedlichen Ergebnissen kommen. Die verwendeten Bauelemente unterliegen gewissen Toleranzen, so dass z.B. Spannungsabfälle an den Dioden oder Transistoren leicht unterschiedlich sein können und so zu unterschiedlichen Ergebnissen führen können. Nicht umsonst werden Messgeräte regelmäßig kalibriert und verwenden besonders hochwertige Bauelemente, aber selbst dann wird man nie zwei identische Ergebnisse bekommen.

Zudem können Timings mit günstiger Hardware nicht genau getestet werden. Hardware, die auf 5 ns genau misst, erhöht die Kosten eines Testers locker auf über 1000 Euro (sehr vorsichtig gerechnet und unter Umständen auch sehr viel mehr, wenn nur wenige Geräte verkauft werden).

Wenn Sie 40 Jahre alte Chips auf ihre technischen Spezifikationen hin testen, werden Sie vermutlich nicht viele Chips finden, die diese noch erfüllen, d.h. in Ordnung sind.

Fazit:

Einen 100% sicheren Tester gibt es nicht, auch nicht mit sehr viel teurerer Hardware, die im Ergebnis vielleicht einen von hundert Chips mehr korrekt erkennt. Wird aber ein Chip als fehlerhaft gemeldet, so ist dieser mit sehr hoher Wahrscheinlichkeit auch tatsächlich defekt.

7.2 Testen von “non-volatile static RAM” (NVRAM) (“Zeropower” RAMs)

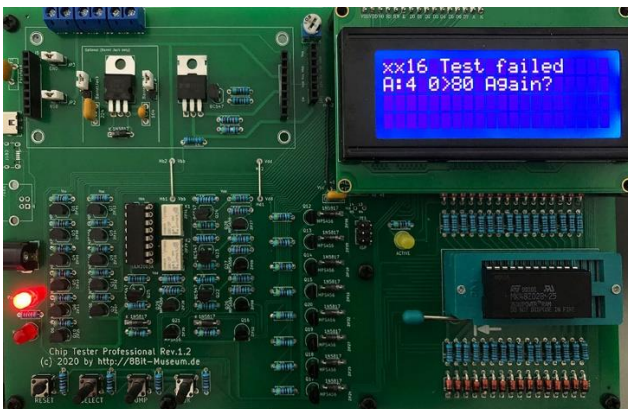
Beim Testen von NVRAMs oder “Zeropower” RAMs, wie z.B. dem M48Z02 oder M48Z12 ist die Spannungsversorgung unbedingt zu beachten.

Durch die Schaltung von Vcc durch einen MOSFET und einen PNP-Transistor ist ein Spannungsabfall von ca. 0,3V zu erwarten. Wird der Tester per USB mit Spannung versorgt (i.d.R. knapp unter 5V), wird sich Vcc letztendlich im Bereich von 4,5V bis 4,7V bewegen. Ein Blick in das Datenblatt zeigt, dass bei einem M48Z02 im Bereich von 4,5V – 4,75V und bei einem M48Z12 im Bereich von 4,2V – 4,5V, der Power-Failure anspricht und der Chip die Ausgänge in den Tristate-Mode schaltet. Damit wird der Chip dann vermutlich als fehlerhaft getestet.

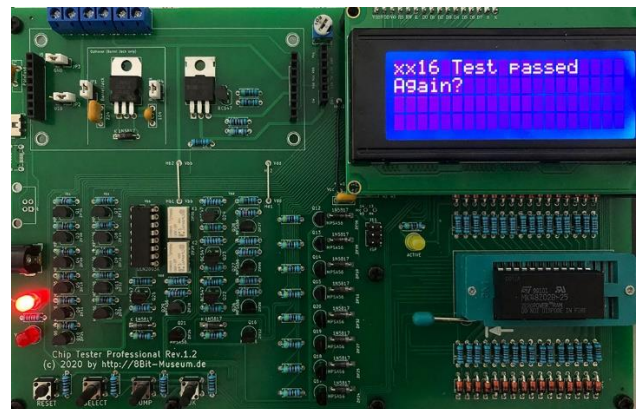
- Automatic power-fail chip deselect and WRITE protection
- WRITE protect voltages (V_{PFD} = power-fail deselect voltage):
 - M48Z02: $V_{CC} = 4.75$ to 5.5 V;
 4.5 V $\leq V_{PFD} \leq 4.75$ V
 - M48Z12: $V_{CC} = 4.5$ to 5.5 V;
 4.2 V $\leq V_{PFD} \leq 4.5$ V

Auszug aus dem Datenblatt des M48Z02 und M48Z12

In diesem Fall sollte der Tester über den Hohlstecker mit 7V bis 9V versorgt werden. Der verwendete Linearregler liefert relativ exakt 5V, was ein Vcc von knapp 4.7V – 4.8V bedeutet. Ein M48Z02 (und erst recht der M48Z12) sollte sich damit fehlerfrei testen lassen.



Spannungsversorgung per USB



Spannungsversorgung per Hohlstecker (9V)

7.3 Erkennen von Fehlern (Logik- und Speicherbausteine)

Grundsätzlich gelten die in Abs. 7.1 beschriebenen Probleme auch für Logikchips. Die Logikchips sind sogar noch etwas komplizierter zu testen, denn für jeden einzelnen Chip muss ein eigener Test vorhanden sein.

Die meisten Tester für Logikchips testen nur die reine Logik, so auch der Retro Chip Tester Pro. Es können aber auch weitere (seltene) Fehler auftreten, die so nicht erkannt werden können (insbesondere bei alten bipolaren Logikbausteinen):

1. ungültige Signalpegel aufgrund von Chipalterung (z.B. ein vorhandener Gleichspannungsoffset bei einem logischen LOW, den der Tester als logisch HIGH erkennt),
2. defekte Eingangstreiber, die aufgrund des Alters mehr Strom benötigen,
3. defekte Ausgangstreiber, die nicht mehr ausreichend Strom liefern (i.d.R. sollten die Ausgangstreiber ausreichend Strom liefern um eine bis zwei TTL Stufen anzusteuern),
4. weitere Probleme.

7.3.1 Ungültige Signalpegel

Der Chip Tester kann technisch bedingt folgende Pegel erkennen:

ATmega2560	gem. Datenblatt	bei 5V
Input Low Voltage	max. 0.3 V _{CC}	max.1.5 V
Input High Voltage	min. 0.6 V _{CC}	min. 3 V

TTL Chips (nicht CMOS) besitzen folgende Charakteristik:

TTL 5V Pegel	Eingang	Ausgang
Low Voltage	max. 0.8 V	max. 0.4 V
High Voltage	min. 2.0 V	min. 2.4 V

Sollte z.B. ein Baustein aufgrund seines Alters tatsächlich nur noch 2.4V für ein logisch HIGH am Ausgang liefern, wird dieses Signal u.U. vom ATmega2560 nicht als solches erkannt.

Das Problem haben alle Tester, die nicht explizit auf den verwendeten Logik- Pegel eingestellt werden. Der beliebte China-Programmer TL866 (altes Modell) verwendet als Bustreiber einen 74HC373D, erkennt also die folgenden Logik- Pegel:

Symbol	Parameter	Conditions	T _{amb} = 25 °C		
			Min	Typ	Max
74HC373					
V _{IH}	HIGH-level input voltage	V _{CC} = 2.0 V	1.5	1.2	-
		V _{CC} = 4.5 V	3.15	2.4	-
		V _{CC} = 6.0 V	4.2	3.2	-
V _{IL}	LOW-level input voltage	V _{CC} = 2.0 V	-	0.8	0.5
		V _{CC} = 4.5 V	-	2.1	1.35
		V _{CC} = 6.0 V	-	2.8	1.8

Abbildung 7.2: Logik- Pegel des 74HC373

Ein IC, der nur noch 2.4V für ein logisch HIGH liefert, würde vom TL866 noch als solches erkannt werden, der Chip Tester wird dieses Signal vermutlich als logisch LOW erkennen. Umgekehrt erkennt der TL866 auch noch bei über 1V ein LOW, was nicht mehr korrekt ist.

In der Praxis sollte ein TTL IC mindestens eine Spannung von 3.5V liefern können (2.4V schreibt der Standard vor), aufgrund der geringen Last vermutlich eher etwas mehr. Gealterte ICs können aber eine geringere Spannung liefern, so dass der Chip Tester das HIGH Signal nicht mehr sicher erkennt. Bei mehrfachen Testen schlägt der Test dann manchmal fehl, manchmal ist er positiv (Anzeige „L“ mit einem Pfeil nach oben = erwartet wurde ein HIGH, gelesen wurde aber ein LOW).

Der TL866 II verzichtet auf die Treiber und gibt die Signale direkt auf die Ports des verwendeten PIC24FJ256GB110. Damit werden folgende Logik-Pegel erkannt.

PIC24FJ256GB110	gem. Datenblatt	bei 5V
Input Low Voltage (ST)	max. 0.2 Vcc	max. 1 V
Input High Voltage (ST)	min. 0.8 Vcc	min. 4 V

Somit kann das Ergebnis beim Testen von gealterten Chips mit einem TL866 II nochmals unterschiedlich sein.

Da die Logik-Pegel von keinem Tester genau eingehalten werden und grundsätzlich nur die Logik getestet wird, bedeutet ein positiver Test, dass der Chip „höchstwahrscheinlich“ in Ordnung ist, während ein fehlgeschlagener Test oder ein Test, der „hin und wieder“ fehlschlägt auf einen „höchstwahrscheinlich“ defekten Chip hindeutet. Je nach Situation ist die Wahrscheinlichkeit für ein korrektes Ergebnis mal bei dem einen, mal bei dem anderen Tester etwas besser.

Empfehlung: Sollte eine Prüfung auf einem Tester – egal welchem – aufgrund der Signalpegel fehlschlagen, auf einem anderen aber nicht, dann sollte der Chip ggf. trotzdem aussortiert werden, weil ein Signalpegel offensichtlich nicht mehr optimal sind (auch wenn evtl. 2.4V noch dem Standard entsprechen, wird ein solcher Chip vermutlich nicht mehr lange funktionieren).

7.3.2 Defekte Eingangs- bzw. Ausgangstreiber

Die bipolaren Bausteine sind stromgesteuert, d.h. es muss ein bestimmter Strom zur Verfügung stehen, damit ein Signal sicher erkannt wird. Ein Beispiel hierfür ist in Abs. 3.7.4 beschrieben (der Chip Tester verfügt zum Schutz über eine Strombegrenzung). Umgekehrt belastet der Chip Tester die Ausgänge eines Logikbaustein nur sehr wenig, so dass auch Bausteine, die nicht mehr in der Lage sind ein oder zwei TTL-Stufen anzusteuern, noch fehlerfrei im Tester funktionieren können, aber in einer realen Schaltung keinen weiteren Chip mehr ansteuern können.

7.3.3 Timing-Probleme

Wird ein Speicherbaustein mehrfach getestet und tritt der Fehler an unterschiedlichen Speicherzellen auf, dann ist es durchaus möglich, dass der Speicher zu langsam für den Test ist (d.h. die Zugriffszeit über die Jahre größer geworden ist). Diese Speicherbausteine können in einem Gerät durchaus noch funktionieren. Erhöhte Zugriffszeiten fallen i.d.R. in einem Zielsystem nicht auf, jedoch im Tester werden diese erkannt.

Es ist auch möglich, dass der RCT einen Speicher als funktionierend testet, dieser Speicher aber in einem Gerät nicht korrekt funktioniert. Es sind dann die Speichermatrix, der Adressdecoder, die Ausgabetreiber etc. grundsätzlich in Ordnung, aber der Speicher kommt mit den schnellen Zugriffszeiten im Zielsystem nicht klar. Hier sollte zunächst geprüft werden, ob das IC überhaupt die richtigen Zugriffszeiten für dieses System hat. Der RCT testet Speicher immer mit konservativen Timings, wodurch es (in seltenen Fällen) auch vorkommen kann, dass Speicher mit starken Alterungseffekten,

die nicht mehr ihre Nennzugriffsgeschwindigkeit besitzen, aber die Grenzwerte im RCT erfüllen, als fehlerfrei getestet werden.

7.3.4 „Falsche“ Ergebnisse

Symptom: Es wird ein IC als OK getestet. In einer realen Schaltung funktioniert er aber nicht.

Das kann mehrere Gründe haben. Der Tester hat festgestellt, dass z.B. bei einem Speicher alle Speicherzellen in Ordnung sind, auch die Ein- und Ausgabetreiber sind ok und ebenso der Adressdecoder. Was aber zu beachten ist, dass der IC in einer (fast) perfekten Umgebung getestet wurde. Die Spannungsversorgung beträgt 5V, ist stabil und es gibt quasi keine Störungen durch andere Komponenten. Auch die Signale sind fast perfekt und besitzen saubere Flanken. In einem realen Gerät, wie einem C64, kann sich der IC durchaus noch fehlerhaft verhalten. Im realen Gerät kann auch eine höhere Temperatur eine Rolle spielen oder der Fehler liegt doch woanders, z.B. an einem defekten Bustreiber.

Symptom: Es wird ein IC als FAIL getestet. In einer realen Schaltung funktioniert er aber.

Es gibt ICs, die aufgrund von Alterungseffekten ihre Spezifikation nicht mehr einhalten (z.B. langsamere Zugriffszeiten besitzen als angegeben) und deshalb im Tester als fehlerhaft ausgewiesen werden. In einem realen System können diese noch funktionieren, wenn dort die Zugriffe langsamer sein sollten.

Oft wird empfohlen einen Speicher in einem lauffähigen Computer einzusetzen und so zu testen, ob er funktionsfähig ist. Das ist nicht aussagekräftig. Ein häufiges Problem ist ein defekter Adressdecoder (Beispiel siehe Kapitel 3.1.4, defekte Adressleitung A4). Wenn z.B. bei einem 64k Speicher die oberste Adressleitung auf einem Signalpegel „festhängt“, werden die 64k in zwei 32k Hälften „gespiegelt“. Ein normaler Pattern Test, den einige Computer eingebaut haben, entdecken diesen Fehler nicht. Wird beim Testen des Rechners nur die eine Hälfte verwendet, fällt der Fehler nicht auf, der Computer scheint fehlerfrei zu laufen.

In diesen Fällen hat der Tester mit seinem Ergebnis recht, das beobachtete Verhalten ist aber u.U. ein anderes.

7.3.5 Weitere Probleme

Es gibt weitere Probleme, die in einer realen Schaltung auftreten können, aber nicht getestet werden können, darunter thermische Probleme, die nur dann auftreten, wenn ein Chip z.B. mit hohen Frequenzen betrieben wird und Fehler erst nach einiger Zeit, wenn er sich erwärmt hat, auftreten. Hier kann das Testergebnis OK sein, im realen System treten dann aber noch einiger Zeit Fehler auf.

Obiges gilt prinzipiell für alle bezahlbaren Tester für Logikchips. Dennoch ist die Erkennungsrate sehr hoch und Fehler können i.d.R. mit einer sehr hohen Wahrscheinlichkeit erkannt werden.

7.4 "In circuit" Tests

"In circuit"-Tests können mit einigen (Logik-)ICs funktionieren, und es gibt einige Berichte, dass es gut mit dem RCT funktioniert hat. Wenn eingelötete ICs getestet werden, muss der Test unbedingt so durchgeführt werden, dass die Platine mit dem zu testenden IC ohne Stromversorgung ist. In den meisten Fällen sollten Sie auch sicherstellen, dass der Vcc-Pin des ICs nicht mit der Platine verbunden ist, da diese sonst über den RCT mit Strom versorgt wird.

Vorsicht bei den Testergebnissen: Es soll z.B. ein NAND-Gatter getestet werden, bei dem beide Eingänge verbunden sind (das Gate wird somit als Inverter eingesetzt). Das ist eine sehr häufig vorkommende Schaltung auf alten Platinen. Wenn der RCT den IC überprüft, schlägt der Test fehl, da sich das Gate wie ein Inverter und nicht wie ein NAND-Gatter verhält.

„In circuit“ Tests sind nicht wirklich zuverlässig möglich. Es *kann* funktionieren, aber in vielen Fällen nicht, was stark von der Schaltung abhängt. Heutige ICs bieten dafür eine spezielle Funktion (Boundary Scan und Test) aber ICs aus den 80er Jahren bieten das nicht.

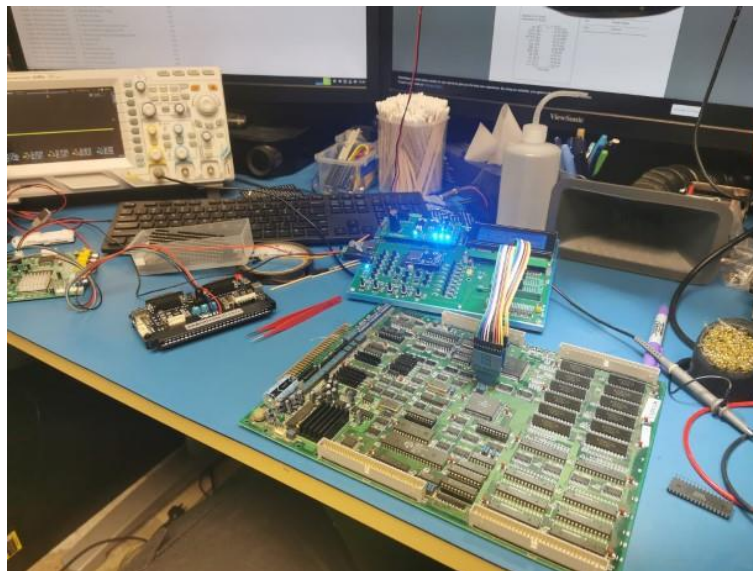


Abbildung 7.3: RCT "in circuit" Tests (Bild von Darren Jones)

Der RCT kann ICs "in circuit" genauso gut oder schlecht testen wie andere Tester. Aber die Wahrscheinlichkeit, dass ein Test bei einem fehlerfreien IC fehlschlägt, ist sehr hoch.

7.5 Verwenden einer SD-Karte

Damit der RCT eine SD-Karte erkennt müssen folgende Punkte erfüllt sein:

- In der Konfiguration muss die Unterstützung für SD-Karten eingeschaltet sein ("**SD Card: 1**", siehe Kap. 5) und der Micro-SD Kartenadapter muss korrekt mit dem Chip-Tester verbunden sein.
- Die Speicherkarte muss FAT16 oder FAT32 formatiert sein. Falls sie nicht erkannt wird, ggf. einmal mit dem SD Memory Card Formatter der SD Association formatieren.
Download unter: <https://www.sdcard.org/downloads/formatter/>
- Es wird nur eine Primäre Partition unterstützt, keine logischen Laufwerke.

Da der Tester über keine Echtzeituhr verfügt, wird standardmäßig kein Dateidatum gesetzt. Es gibt jedoch die Möglichkeit ein Datum mit Uhrzeit vorzugeben, indem man auf der SD-Karte eine Datei `datetime.txt` anlegt. Wird der Tester eingeschaltet (bzw. nach einem Reset), werden Datum und Uhrzeit dieser Datei beim Anlegen von weiteren Dateien verwendet.

8 Anhang: PAL Analyse und Synthese mit dem RCT

In Kapitel 3.5 wird beschrieben, wie ein PAL oder GAL ausgelesen werden kann. Die zugrunde liegenden logischen Gleichungen lassen sich mit sehr geringem Aufwand rekonstruieren, wenn alle möglichen Eingabekombinationen durchprobiert werden, die Ausgaben protokolliert und anschließend analysiert werden. Auf diese Weise kann eine Kopie erstellt werden, die logisch identisch mit dem Original ist.

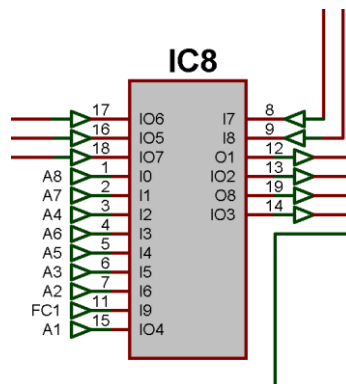
Dieses Kapitel beschreibt, wie dabei vorgegangen werden kann. Der folgende Weg wurde mit Hilfe von Christophe Escurat dokumentiert. Vermutlich gibt es auch noch einfachere Lösungen.

Folgende Software wird benötigt:

- **PALASM 1.5** wird verwendet, um eine JEDEC Datei zu erstellen
<http://www.pldworld.com/otherplds/palasm/-engr.uky.edu/melham01/ee481/software.htm>
 Da es ziemlich alt ist und MS-DOS benötigt, verwenden wir **DOSBOX**, um es auszuführen.
- **DOSBOX** wird verwendet um PALASM auszuführen
<https://www.dosbox.com/download.php?main=1>
- **Simple Solver** reduziert die Logikgleichungen
<https://www.simplesolverlogic.com/>

Schritt 1: Analyse

Zunächst muss spezifiziert werden, welche Pins Eingabe- bzw. Ausgabe-Pins sind. Dazu ist es am einfachsten, die Schaltpläne zu prüfen oder, falls nicht verfügbar, ein kleines Reverse Engineering durchzuführen. In diesem Beispiel analysieren wir einen PAL16L8.



Die I/O-Pins sind im Menü „PAL (20 Pins)“ wie folgt zu spezifizieren:

```
12345678901234567890
IIIIIIIIIIIGIOOOIIIIIOV
```

Schritt 2: Speichern des PAL Inhalts

Der TI16L8-25 in unserem Beispiel ist ein PAL16L8 in einem PLCC-Gehäuse. Wir verwenden einen PLCC-20-zu-DIP-20-Adapter, um den PAL im Retro Chip Tester auszulesen. Wählen Sie das Menü „Selected (PAL) / PAL 20pin“ und stellen Sie sicher, dass der SD-Kartenadapter installiert und in der Konfiguration des RCT aktiviert ist. Stellen Sie die PIN-Konfiguration auf das oben erwähnte Pattern ein. Halten Sie die OK-Taste zwei Sekunden gedrückt, um den Vorgang zu starten.



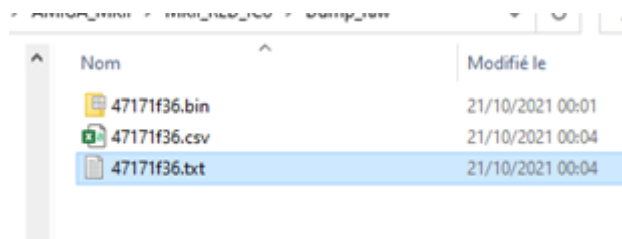
Der Inhalt wird nun auf die SD-Karte geschrieben. Dies kann je nach Anzahl der Kombinationen eine Weile dauern. Zeit für einen Kaffee. Der RCT prüft 4000 = 16384 Kombinationen durch.



Das Speichern erfolgt in zwei Schritten: Im ersten Schritt erstellt der RCT eine binäre Dump-Datei, im zweiten Schritt wird diese Datei in eine CSV-Datei konvertiert, die von Excel oder einem Texteditor geöffnet werden kann.

Schritt 3: Prüfen der Dump-Datei

Nach einiger Zeit (es kann mehrere Minuten dauern) ist die Konvertierung abgeschlossen. Überprüfen Sie nun den Inhalt der SD-Karte. Kopieren Sie die BIN- und CSV-Datei auf Ihren Computer, erstellen Sie eine Kopie der CSV-Datei und benennen Sie diese in TXT um.



Um die Textdatei mit einem anderen Programm verwenden zu können, müssen wir einige Änderungen vornehmen. Öffnen Sie die Datei mit einem Texteditor und ändern Sie die erste Zeile.

```
47171f36.txt - Bloc-notes
Fichier  Edition  Format  Affichage  Aide
18;17;16;15;11; 9; 8; 7; 6; 5; 4; 3; 2; 1;-;19;14;13;12
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0;-; 0; 1; 0; 0
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1;-; 0; 1; 1; 1
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 0;-; 0; 1; 1; 1
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 1;-; 0; 1; 1; 1
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 0; 0;-; 0; 1; 1; 1
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 1; 0;-; 0; 1; 1; 1
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 1; 1;-; 0; 1; 1; 1
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 0; 0;-; 0; 1; 1; 1
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 0; 0; 1;-; 0; 1; 1; 1
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 0; 1; 0;-; 0; 1; 1; 1
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 1; 0; 0;-; 0; 1; 1; 1
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 1; 0; 1;-; 0; 1; 1; 1
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 1; 1; 0;-; 0; 1; 1; 1
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 1; 1; 0;-; 0; 1; 1; 1
Ln 1, Col 1    100%    Windows (CRLF)    UTF-8
```

Nehmen sie folgende Änderungen **nur an der ersten Zeile** vor:

- Fügen Sie den Nummern der I/O-Pins das Präfix „io“ hinzu.
- Entfernen sie alle „;“ (Semikolon), sodass zwischen den Werten nur ein Leerzeichen bleibt.
- Ersetzen Sie das „-“ (Minus) durch ein „:“ (Doppelpunkt).

Die geänderte Zeile sieht dann folgendermaßen aus:

47171f36.txt - Bloc-notes

Fichier Edition Format Affichage Aide

```

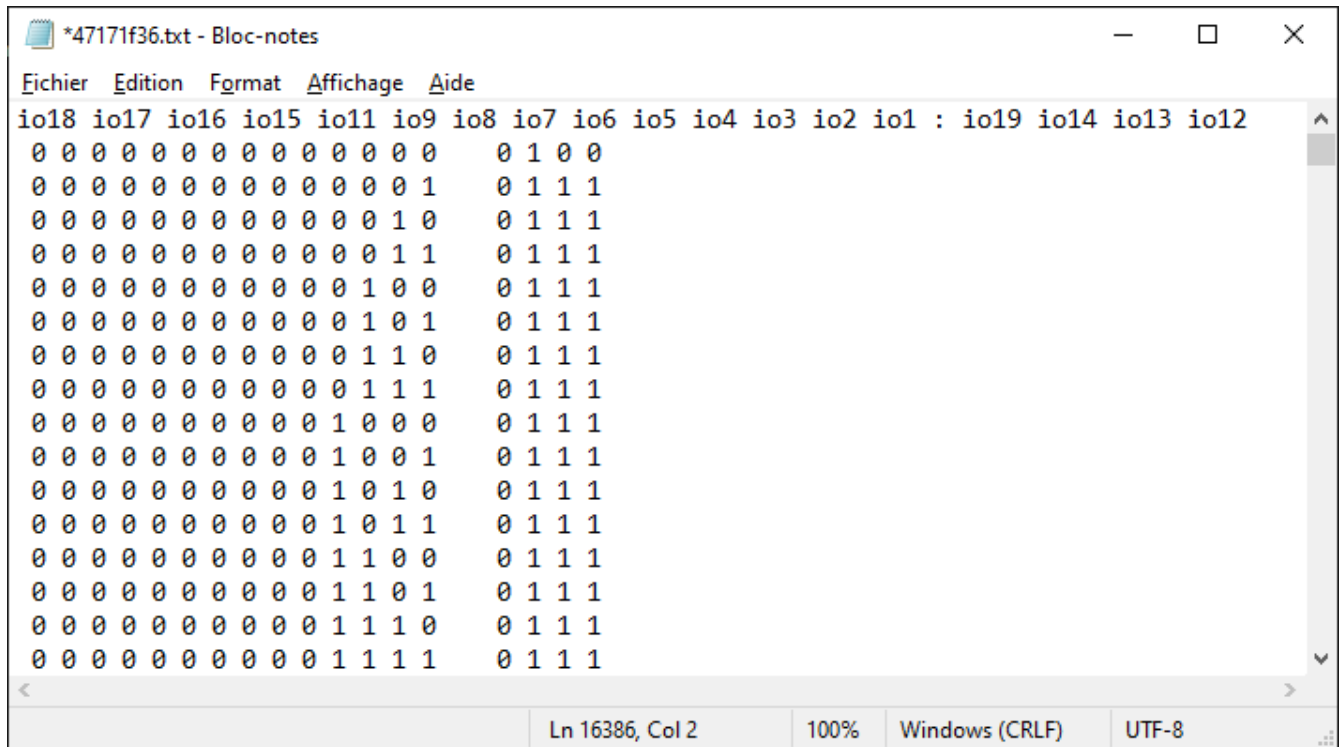
io18 io17 io16 io15 io11 io9 io8 io7 io6 io5 io4 io3 io2 io1 : io19 io14 io13 io12
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; -; 0; 1; 0; 0
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; -; 0; 1; 1; 1
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 0; -; 0; 1; 1; 1
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 1; -; 0; 1; 1; 1
0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 1; 0; 0; -; 0; 1; 1; 1

```

Ln 1, Col 83 100% Windows (CRLF) UTF-8

Ändern Sie die restlichen Zeilen wie folgt:

- Entfernen sie alle ";" (Semikolon).
- Ersetzen Sie das „-“ (Minus) durch einen Tabulator.

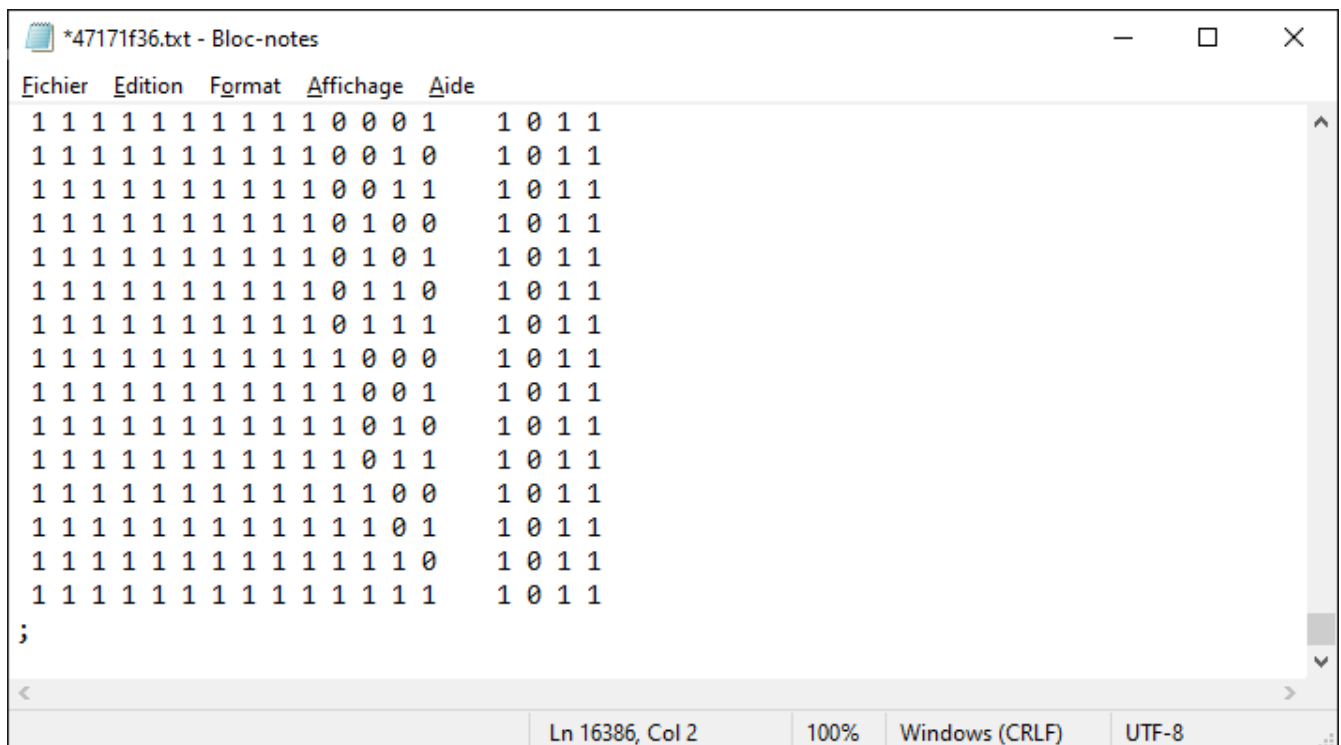


```

*47171f36.txt - Bloc-notes
Fichier Edition Format Affichage Aide
io18 io17 io16 io15 io11 io9 io8 io7 io6 io5 io4 io3 io2 io1 : io19 io14 io13 io12
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1

```

Fügen sie abschließend ein ";" (Semikolon) an das Ende der Datei hinzu und speichern sie diese.



```

*47171f36.txt - Bloc-notes
Fichier Edition Format Affichage Aide
1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 0 1 1
1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
;

```

Schritt 4: Lesen der Wahrheitstabelle und Reduzieren der Logikgleichungen

Wir benötigen die Software PALASM. Da sie ziemlich alt ist läuft sie noch unter MS-DOS. Wir verwenden daher DOSBOX, um diese auszuführen. Die Installation von DOSBOX ist sehr einfach. Verwenden Sie die Konfigurationsdatei, die Sie unter

```
C:\Users\<username>\AppData\Local\DOSBox\dosbox-0.74.conf
```

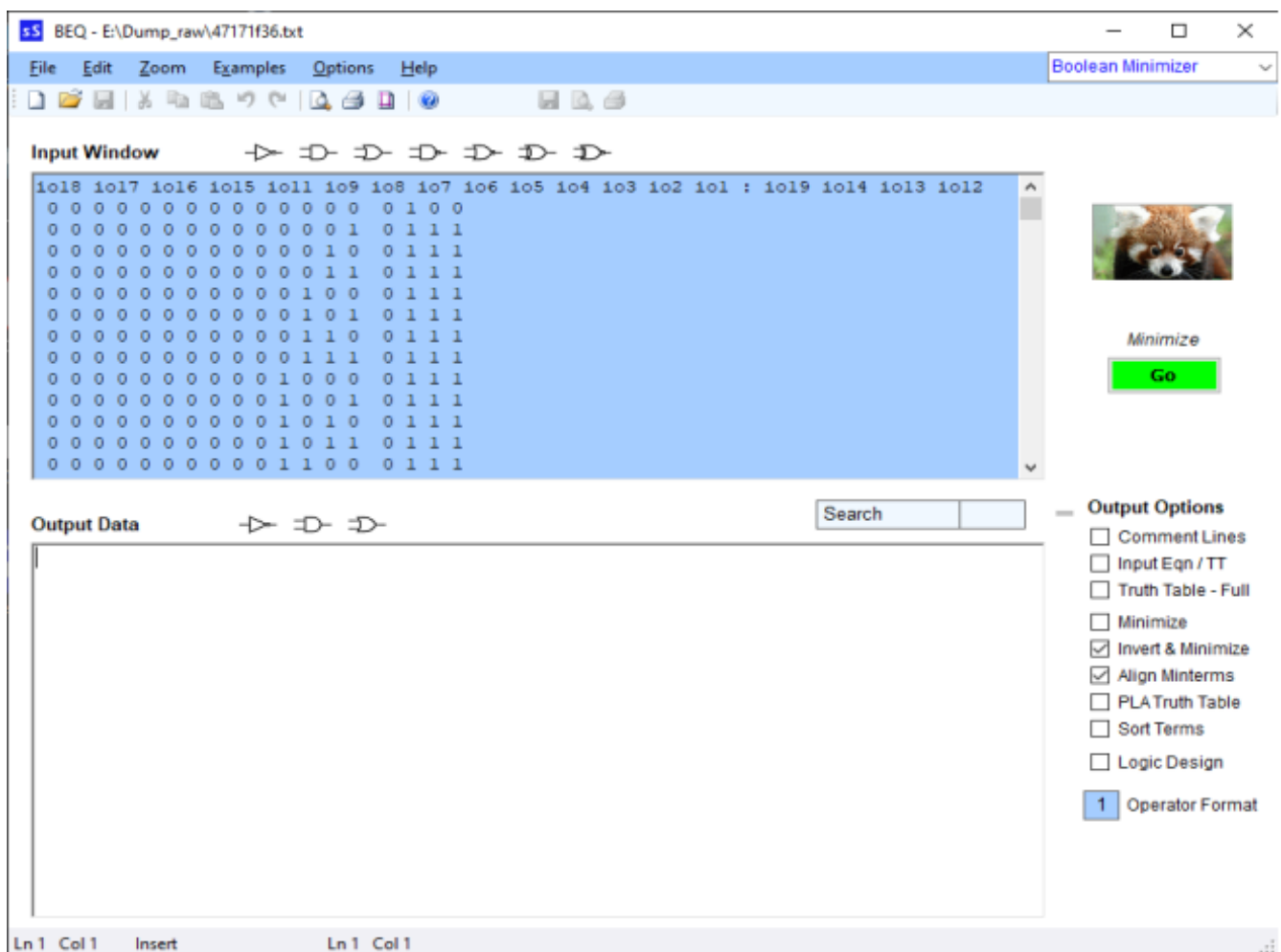
finden und fügen sie folgende Zeilen hinzu:

```
[autoexec]
# Lines in this section will be run at startup.
# You can put your MOUNT lines here.
mount C D:\dosprogs

PATH C:\DOSBOX\PALASM\EXE;C:\;"%PATH%;C:\DOSBOX\PALASM4\EXE"
SET PALASM=C:\PALASM\

C:
CD C:\PALASM
```

Laden Sie nun die Wahrheitstabelle in Simple Solver (die modifizierte Textdatei).



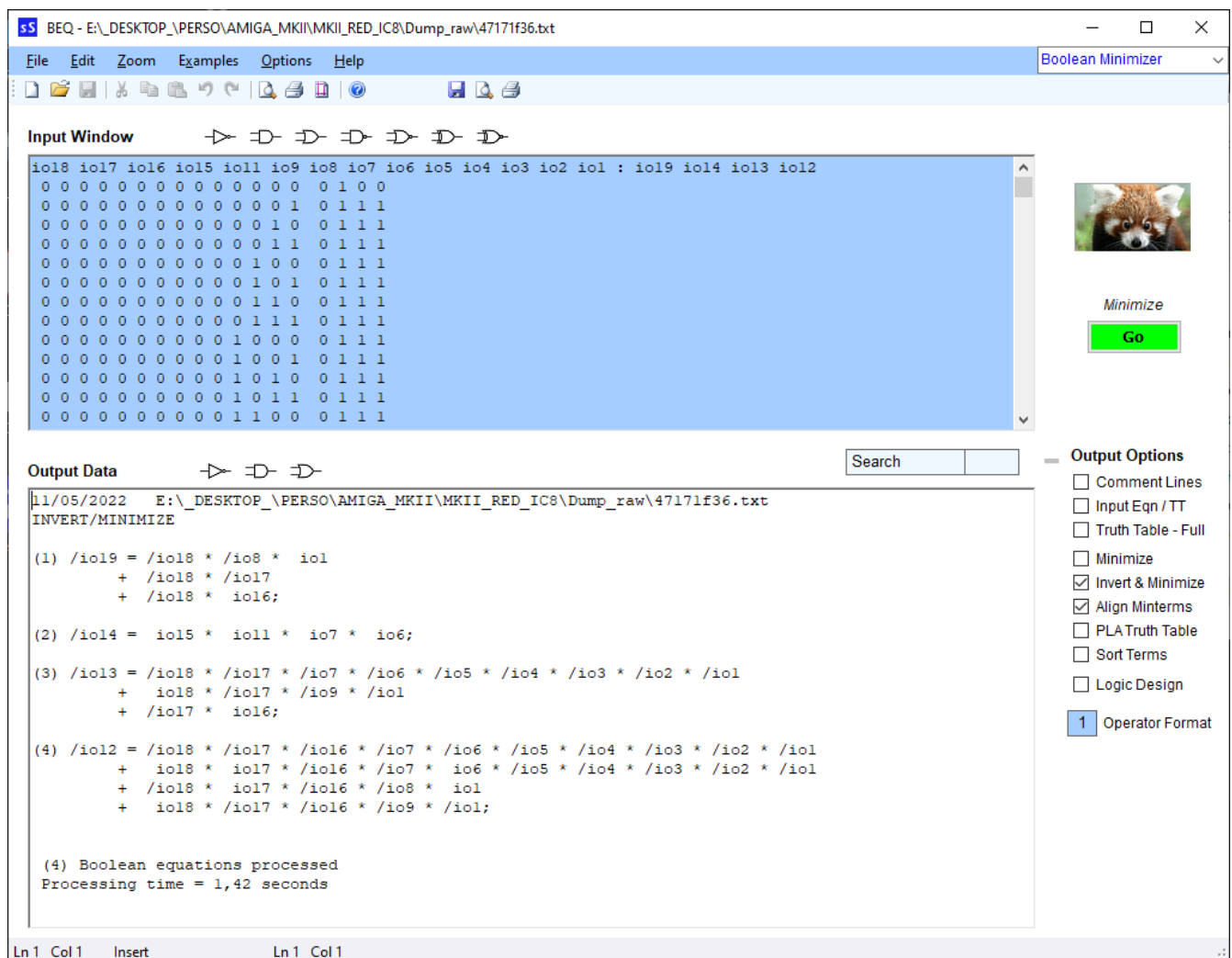
Die Optionen müssen wie folgt geändert werden:

Output Options

- ☐ Comment Lines
- ☐ Input Eqn / TT
- ☐ Truth Table - Full
- ☐ Minimize
- ☒ Invert & Minimize
- ☒ Align Minterms
- ☐ PLA Truth Table
- ☐ Sort Terms
- ☐ Logic Design

1 Operator Format

Drücken Sie GO, um die Daten zu analysieren und die Gleichung zu reduzieren



Schritt 5: Erzeugen der PDS-Datei für PALASM

Erstellen Sie im PALASM-Verzeichnis (z. B. D:\dosprogs\PALASM) eine PDS-Datei und öffnen Sie diese im Texteditor. Kopieren Sie die folgenden Zeilen in die Textdatei. Bitte achten Sie darauf, dass die INPUT- und OUTPUT Pin-Deklarationen identisch mit der originalen PAL Datei sind (siehe oben).

```

;PALASM Design Description

;----- Declaration Segment -----
TITLE    MYPAL1
PATTERN  None
REVISION 0
AUTHOR   Giants
COMPANY
DATE     10/04/2022

CHIP MYPAL1  PALCE16V8                ; PALCE16V8

;----- PIN Declarations -----
pin 1 io1      COMBINATORIAL          ; INPUT
pin 2 io2      COMBINATORIAL          ; INPUT
pin 3 io3      COMBINATORIAL          ; INPUT
pin 4 io4      COMBINATORIAL          ; INPUT
pin 5 io5      COMBINATORIAL          ; INPUT
pin 6 io6      COMBINATORIAL          ; INPUT
pin 7 io7      COMBINATORIAL          ; INPUT
pin 8 io8      COMBINATORIAL          ; INPUT
pin 9 io9      COMBINATORIAL          ; INPUT
pin 11 io11     COMBINATORIAL          ; INPUT
pin 12 io12     COMBINATORIAL          ; OUTPUT
pin 13 io13     COMBINATORIAL          ; OUTPUT
pin 14 io14     COMBINATORIAL          ; OUTPUT
pin 15 io15     COMBINATORIAL          ; INPUT
pin 16 io16     COMBINATORIAL          ; INPUT
pin 17 io17     COMBINATORIAL          ; INPUT
pin 18 io18     COMBINATORIAL          ; INPUT
pin 19 io19     COMBINATORIAL          ; OUTPUT

;----- Boolean Equation Segment -----

EQUATIONS

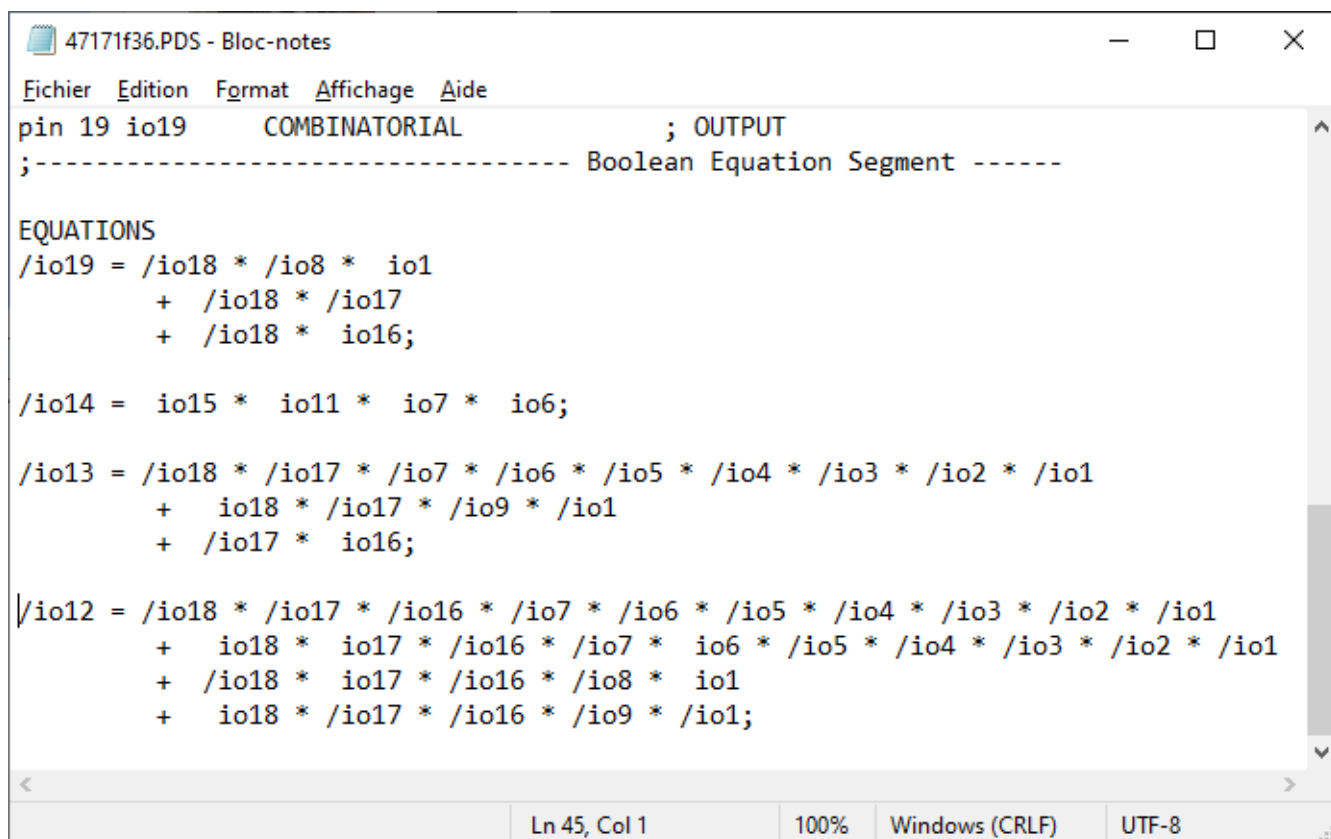
```

Einige wichtige Informationen:

```
CHIP PYPAL1 PALCE16V8
```

PALCE16V8 wird für den GAL16V8 verwendet (siehe PALASM-Dokumentation). Wenn Sie ein anderes „Muster“ verwenden, stellen Sie sicher, dass es gemäß Ihrer Gleichung geändert wird. Es ist notwendig, jeden Pin zu konfigurieren (außer GND und Vcc, Pin 10 und Pin 20 für GAL16V8). Da wir den Namen der Pins in der Textdatei in „io<Nummer>“ geändert haben, müssen wir denselben Namen verwenden.

Im Abschnitt EQUATIONS müssen wir unsere minimierten Gleichungen (nur die Gleichungen) kopieren, die zuvor mit Simple Solver erstellt wurden. Die Datei speichern.



```
47171f36.PDS - Bloc-notes
Fichier Edition Format Affichage Aide
pin 19 io19      COMBINATORIAL      ; OUTPUT
;----- Boolean Equation Segment -----

EQUATIONS
/io19 = /io18 * /io8 * io1
        + /io18 * /io17
        + /io18 * io16;

/io14 = io15 * io11 * io7 * io6;

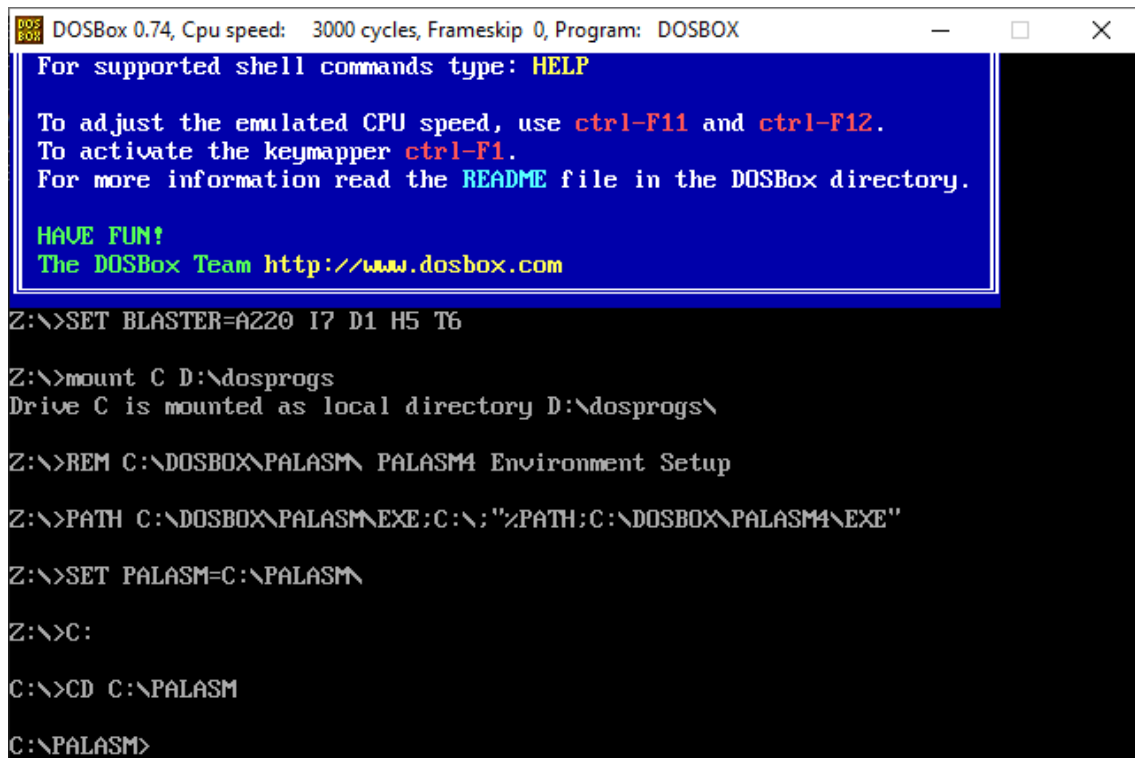
/io13 = /io18 * /io17 * /io7 * /io6 * /io5 * /io4 * /io3 * /io2 * /io1
        + io18 * /io17 * /io9 * /io1
        + /io17 * io16;

/io12 = /io18 * /io17 * /io16 * /io7 * /io6 * /io5 * /io4 * /io3 * /io2 * /io1
        + io18 * io17 * /io16 * /io7 * io6 * /io5 * /io4 * /io3 * /io2 * /io1
        + /io18 * io17 * /io16 * /io8 * io1
        + io18 * /io17 * /io16 * /io9 * /io1;

Ln 45, Col 1    100%    Windows (CRLF)    UTF-8
```

Schritt 6: Kompilieren und Erzeugen der JEDEC-Datei

Starten Sie DOSBOX und führen Sie PALASM aus.



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount C D:\dosprogs
Drive C is mounted as local directory D:\dosprogs\

Z:\>REM C:\DOSBOX\PALASM\ PALASM4 Environment Setup

Z:\>PATH C:\DOSBOX\PALASM\EXE;C:\;"%PATH%;C:\DOSBOX\PALASM4\EXE"

Z:\>SET PALASM=C:\PALASM\

Z:\>C:

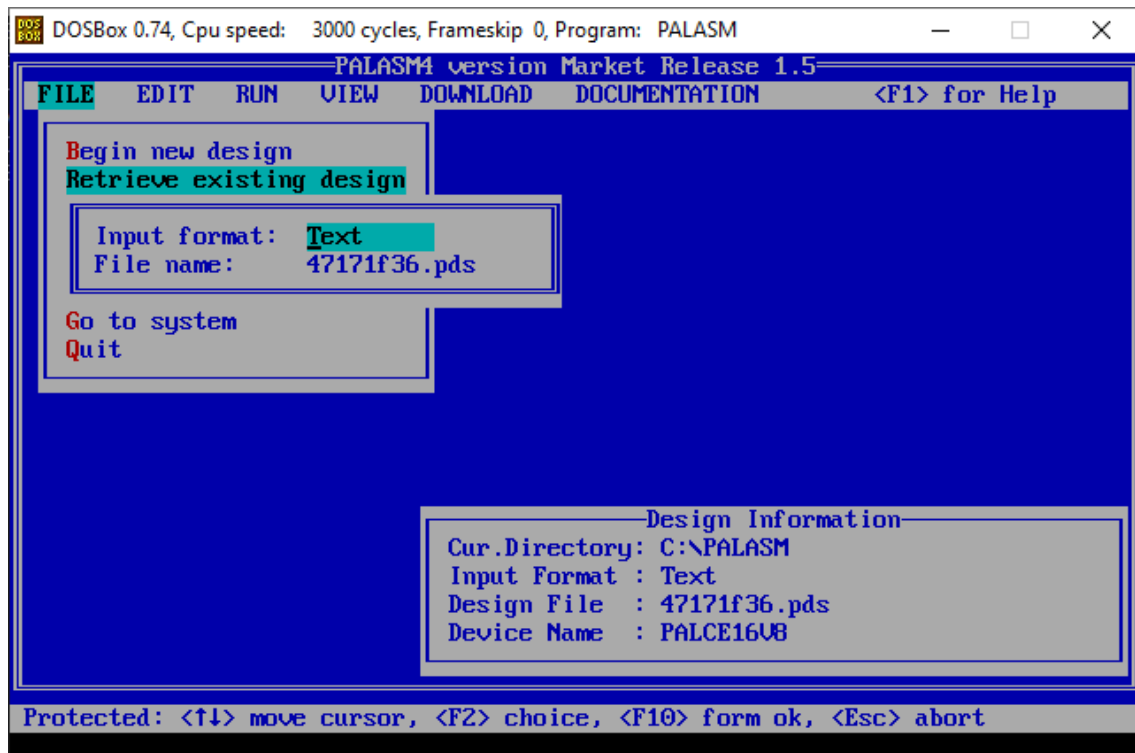
C:\>CD C:\PALASM

C:\PALASM>
```

Laden der PDS-Datei.

FILE → Retrieve existing design → Input Format Text / Filename <your file>.pds

F10 drücken, um die Änderungen zu akzeptieren.



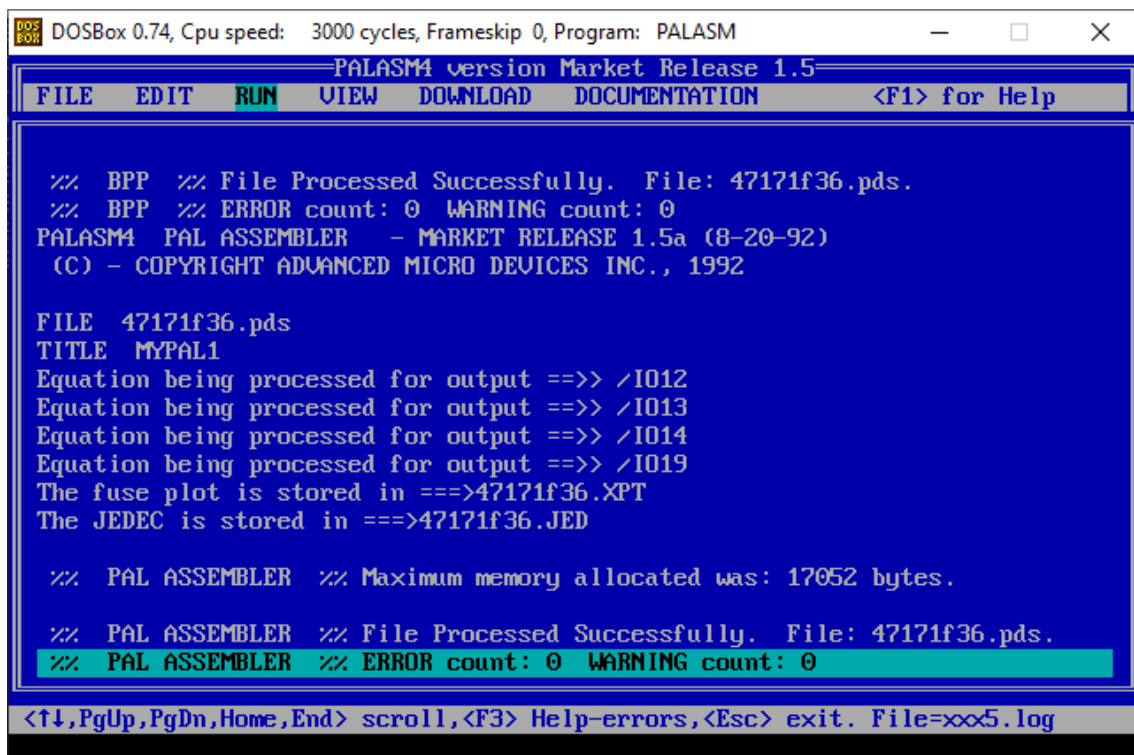
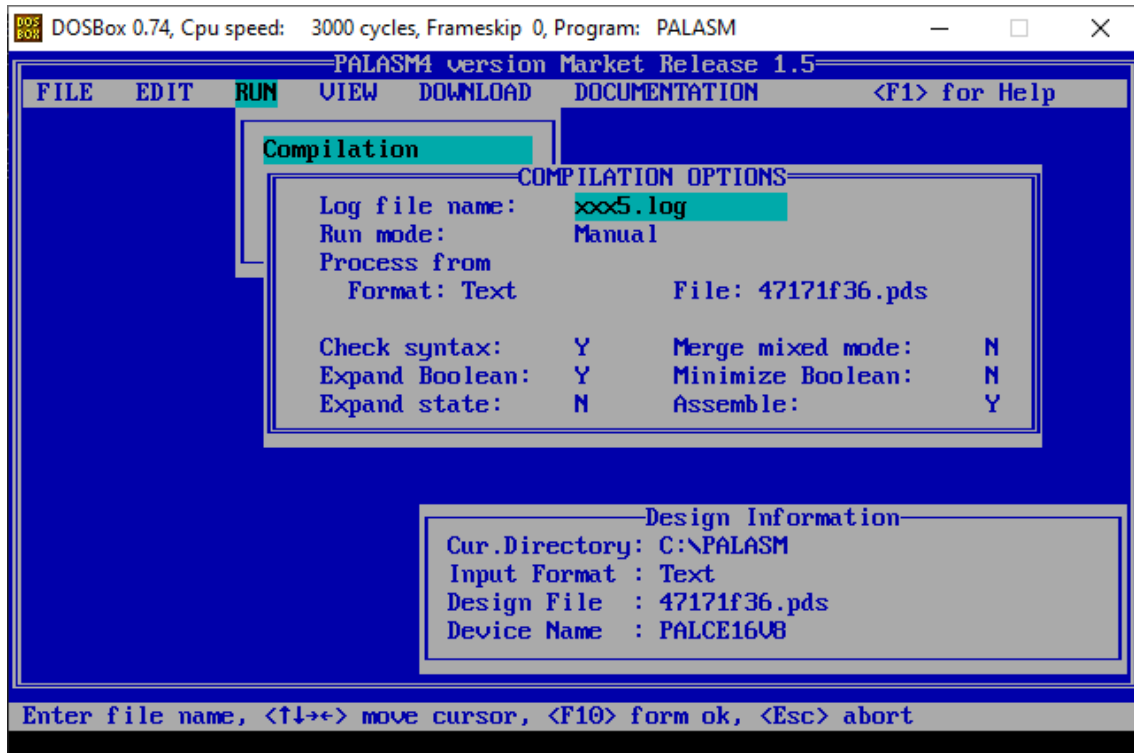
Die Datei kann angezeigt werden mit:

EDIT → TEXT file → [Press ESC Key]

Sie können durch die Datei navigieren und prüfen, ob sie korrekt geladen wurde.

Drücken Sie [F3], um den Editor zu verlassen und die Datei zu kompilieren:

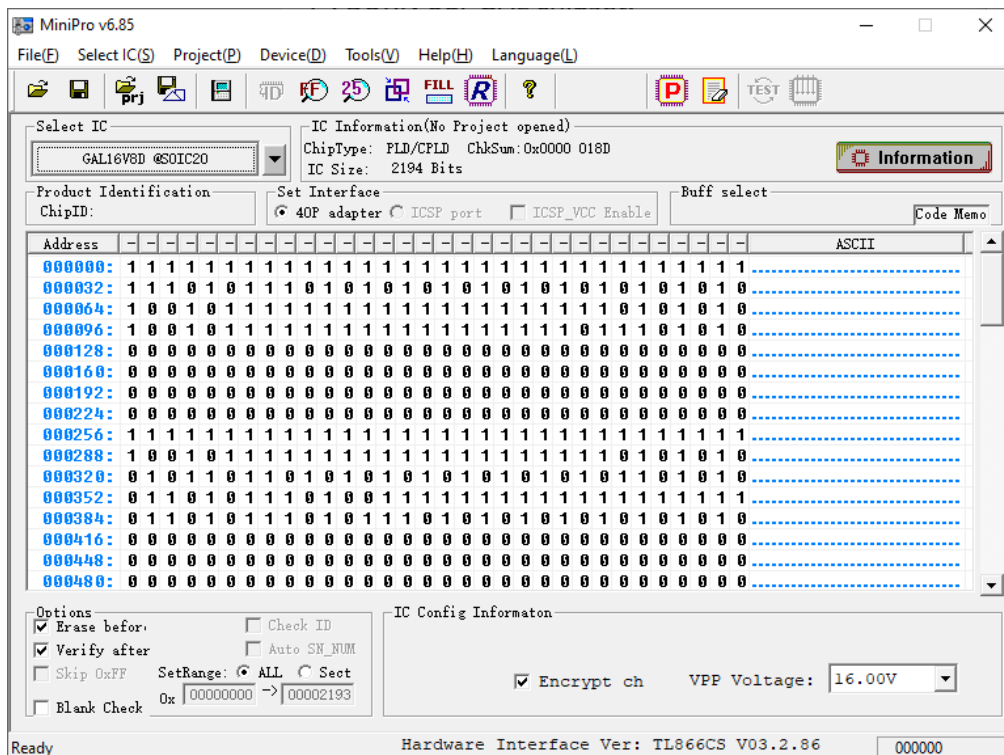
RUN → Compilation → (check the options and press F10)



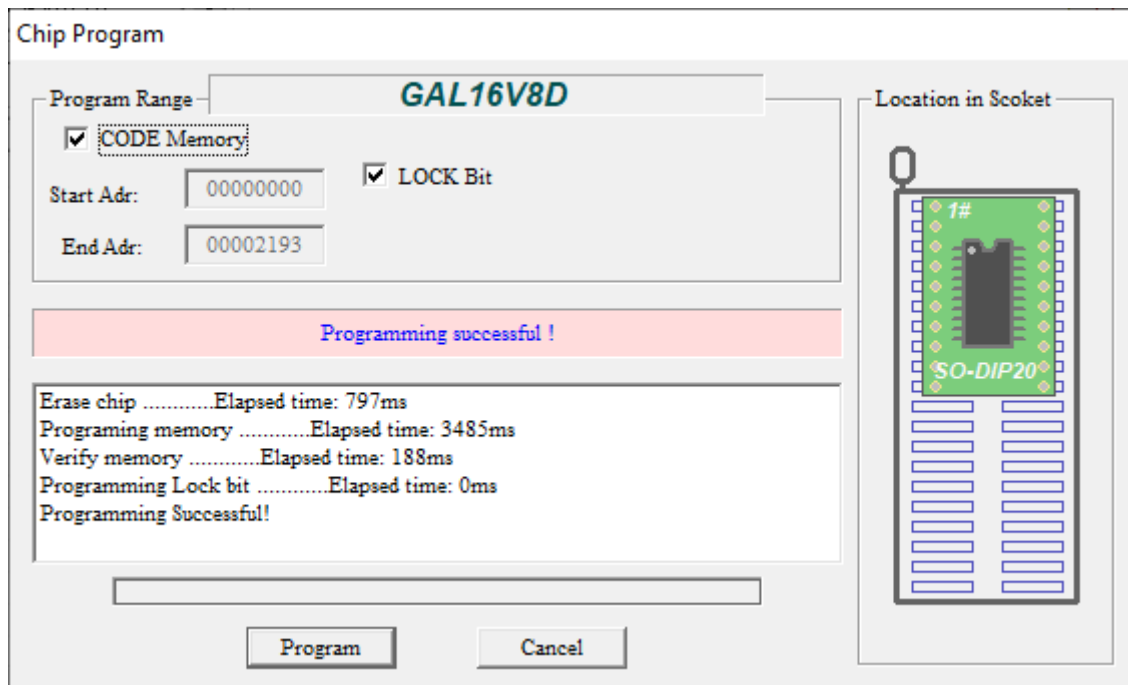
Wenn kein Fehler angezeigt wird, wurde eine JED-Datei erstellt. Beenden Sie PALASM und DOSBOX.

Schritt 7: Brennen des GALs

Setzen Sie den GAL16V8 in einen DIP-20-Adapter und programmieren Sie ihn z.B. mit dem MiniPro-Programmierer. Das GAL16V8D auswählen, JEDEC-Datei laden und programmieren.



Nach einigen Sekunden sollte die GAL programmiert sein.



Herzlichen Glückwunsch. Das war alles.

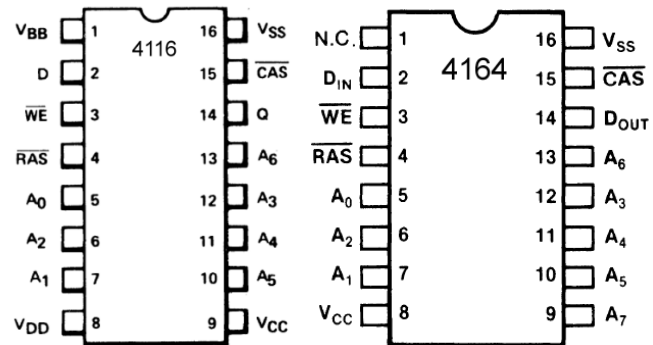
9 Anhang: Ein paar Tipps

Dieses Kapitel enthält einige Tipps, die nichts mit dem RCT zu tun haben, sondern sich auf Speicherchips beziehen. Diese können für den einen oder anderen User nützlich sein.

9.1 Verwenden eines 4164 oder 41256 als 4116

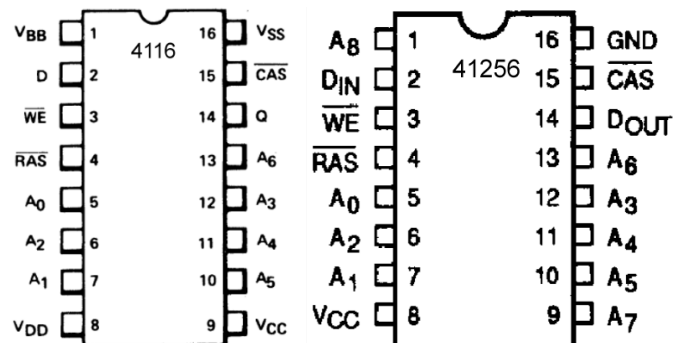
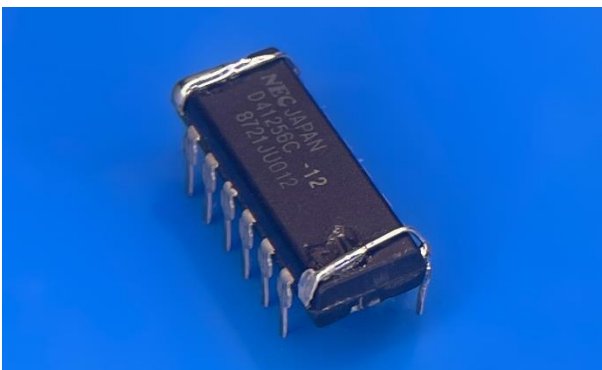
Mit einem 4164 oder 4156 lässt sich sehr leicht ein 4116 ersetzen. Der Vorteil: Der Ersatz benötigt nur eine anstelle der drei Versorgungsspannungen und ist immer noch voll Pin-kompatibel.

Umbau eines 4164 zum 4116:



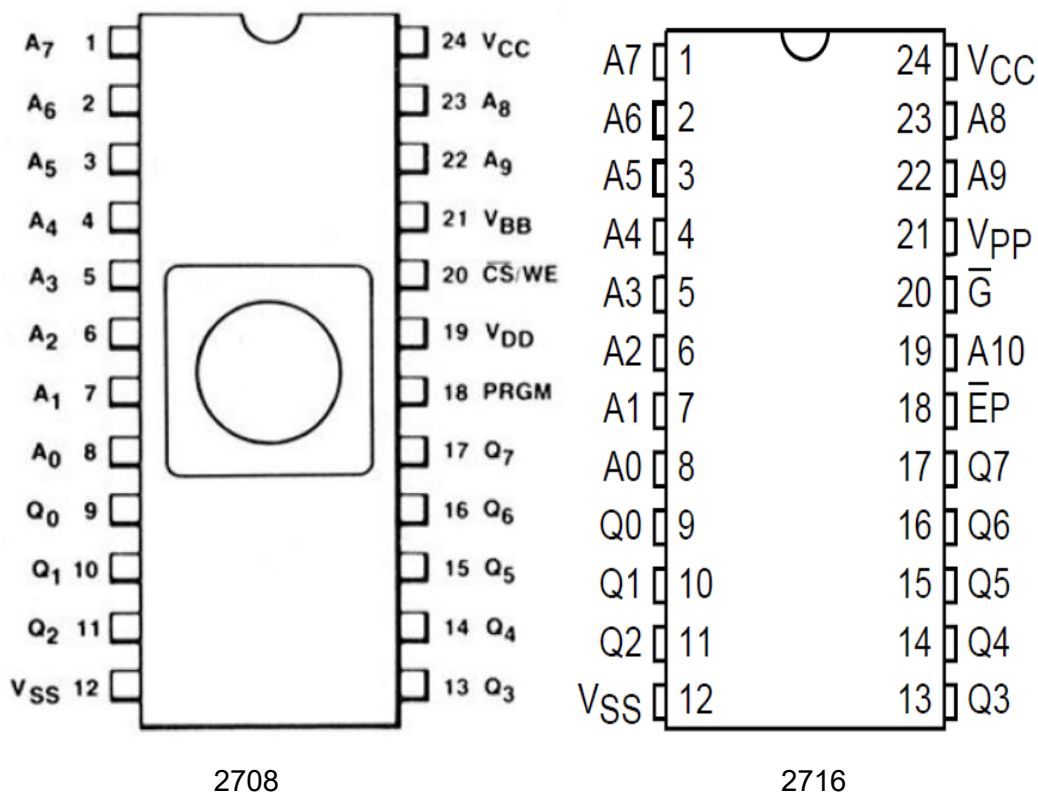
- **Pin 1 entfernen** (der Pin ist nicht belegt, am Sockel liegen -5V an)
- **Pin 8 hochbiegen und mit Pin 9 verbinden** (am Sockel liegen 12V an Pin 8 an, das IC erwartet 5V an Pin 8 und A7 an Pin 9, die beide mit 5V verbunden werden, die am Sockel Pin 9 anliegen)

Umbau eines 41256 zum 4116:



- **Pin 1 hochbiegen und mit Pin 16 verbinden** (das IC erwartet A₈ an Pin 1, welches mit V_{ss} verbunden wird, das an Pin 16 am Sockel anliegt)
- **Pin 8 hochbiegen und mit Pin 9 verbinden** (am Sockel liegen 12V an Pin 8 an; das IC erwartet 5V an Pin 8 und A₇ an Pin 9, die beide mit 5V verbunden werden, was an Pin 9 am Sockel anliegt)

9.2 Verwenden eines 2716 als 2708



Den 2716 wie folgt modifizieren:

- Pin 21 (V_{pp}) hochbiegen und mit Pin 24 (V_{cc}) verbinden
- Pin 19 (A₁₀) hochbiegen und mit V_{ss} oder V_{cc} verbinden (je nachdem, ob die oberen oder unteren 1K angesprochen werden sollen)

Hinweise:

- Pin 20 wird beim 2708 als /CS („chip select“) bezeichnet, ist aber funktionell ein /OE bzw. /G („output enable“).
- Pin 18 wird beim 2708 mit V_{ss} verbunden, beim 2716 ist dieses /CE bzw. /EP („chip enable“ bzw. „enable/program“). Dadurch, dass /CE auf V_{ss} gelegt wird, erreicht der 2716 nie seine minimale Verlustleistung, was aber die Funktion nicht einschränkt.