

Gepufferter DMA- und IM2-fähiger ECB-Bus für GENIE I und II

Helmut Bernhardt

Während bei Verzicht auf DMA- und IM2-Fähigkeit die Steuerung des Datenbuffers zwischen Computer und externem ECB-Bus relativ einfach zu realisieren ist (der Buffer darf nur in Richtung CPU treiben, wenn Daten von Baugruppen gelesen werden, die nicht auf der CPU-Seite des Buffers - also auf dem ECB-Bus - sitzen), sind bei der Einbindung dieser Features einige zusätzliche Aspekte zu berücksichtigen.

Der Interrupt-Mode2 der Z80-CPU setzt voraus, daß der Interrupt-Lieferant beim Interrupt-Acknowledge (M1* und IORQ* beide low) ein 8Bit-Wort auf den Datenbus legt, das die CPU als LSB einer Adresse in der Interrupt-Vektor-Tabelle interpretiert. Dieses Byte muß von einer Interrupt-Quelle vom ECB-Bus durch den Datenbuffer auch zur CPU gelangen. Der Buffer muß beim Interrupt-Acknowledge also auch in Richtung CPU treiben.

Bei Übernahme der Daten-, Adreß- und Steuerbusse durch einen DMA-Controller (DMAC) auf dem ECB-Bus muß dieser auch Zugriff auf RAM, EPROM und alle CPU-seitig des Buffers liegenden Baugruppen haben. Für Adreß- und Steuerleitungen gestaltet sich diese Richtungssteuerung recht einfach. Hier genügt es, wenn durch das low aktive CPU-Signal BUSAK* (nur dann darf der DMAC arbeiten) die Buffer dieser Signale in Richtung zur CPU treiben, so daß der DMAC dort entsprechend adressieren kann. Für die Steuerung des Datenbuffers sind noch weitere Umstände zu berücksichtigen.

Wenn die CPU den Bus kontrolliert, kann beim Schreiben der Buffer immer in Richtung ECB-Bus treiben. Beim Lesen darf er nur dann nicht in Richtung CPU treiben, wenn interne Baugruppen (CPU-seitig des Buffers) adressiert werden. Sonst würde ein von außen gelesenes FFH gegen ein von der intern adressierten Baugruppe ausgegebenes Datenwort arbeiten. Abgesehen davon, daß die gegeneinander arbeitenden Komponenten sich irgendwann gegenseitig zerstören, würde die CPU Fehlinformationen lesen.

Wenn der DMAC ~~(45)~~ den Bus steuert, ist Lesen und Schreiben für die Treiberichtung des Buffers das ganze Gegenteil. Wenn der DMAC Daten einer CPU-seitigen Systemkomponente liest, wobei er dann auch die Leitung RD* low zieht, muß der Buffer in Richtung ECB-Bus treiben. Das darf er aber nur dann, wenn er auf CPU-seitige Komponenten zugreift. Wenn er von einer Baugruppe auf dem ECB-Bus liest, muß der Buffer in Richtung CPU treiben, um dann nicht auf dem ECB-Bus zwei Datenwörter gegeneinander arbeiten zu lassen.

Für die Treiberichtung des Buffers muß also berücksichtigt werden, ob der DMAC oder die CPU den Bus kontrolliert, ob die adressierte Baugruppe auf dem ECB-Bus oder CPU-seitig des Buffers liegt und ob gelesen oder geschrieben wird.

In Tabelle 1 sind die für verschiedene Betriebszustände aus obigen Überlegungen abgeleiteten Treiberichtungen (von der CPU aus gesehen) zusammengestellt. Dabei ist zu berücksichtigen, daß zusätzlich auch noch beim Interrupt-Acknowledge die Richtung "rein" sein muß, weil der GENIE keine eigenen vektorisierten INT-Quellen hat. Diese Tabelle ist Grundlage der Schaltung in Abb.1. Die Richtungssteuerung soll ausschließlich aus Signalen, die von CPU-seitigen Baugruppen zu beziehen sind, erfolgen. Nur dann ist auch gewährleistet, daß auf dem ECB-Bus jede handelsübliche Karte funktioniert. Kommerzielle ECB-Bus-Karten legen kein aktives Signal auf den Bus, das in die Buffersteuerung einbezogen werden kann.

Tabelle 1: Treiberrichtung des ECB-Bus Databuffers

Bus- Steuerung durch	interne Baugruppe adressiert		externe Baugruppe adressiert	
	Read	Write	Read	Write
CPU	raus	raus	rein	raus
DMAC	raus	rein	rein	rein

Der Aufwand dafür ist aber nicht unerheblich und der Anwender dieser Bussteuerung muß sein System genau kennen, um die nötigen Steuersignale an die Schaltung legen zu können. Einen Anhaltspunkt dafür, welche Freigabesignale vorkommen können, mag die Tabelle 2 geben. Darin sind die prinzipiell vorhandenen Baugruppen und einige häufig benutzte Hardware-Erweiterungen und die Punkte, wo deren Freigabesignale abzugreifen sind, zusammengestellt. Diese Punkte sind mit jeweils einem der Punkte A bis H auf der Eingangskarte zu verbinden.

Tabelle 2: Interne Freigabesignale

CPU-Board	Adresse	IC	Typ	Pin	
RAM	4000-FFFF (3900-3BFF)	<7>	37	367	15
ROM1	0000-0FFF		10	3001	20
ROM2	1000-1FFF		11	3002	20
ROM3	2000-2FFF		12	3003	20
ROM4	3000-36FF(37DF)	<7>	13	2716	20
Tastatur	3800-3BFF(38FF)	<7>	7	368	1
Video-RAM	3C00-3FFF		35	32	3
Video-Board					
Cassette 1	Port FF		20	32	11
Cassette 2	Port FE	bleibt unberücksichtigt	#1		
Expansion-Interface	EG3014				
RAM1	8000-BFFF		37	20	10
RAM2	C000-FFFF		37	20	9
Floppy	37E0-37FF		29	139	12
Drucker	37E8 u. Port FD		31	156	9
RS 232 EG3020	Ports F8, F9		37	20	10
EXP1 von RB	37E0-37FF		18	155	2,14
	Port FD		14	32	3
RB.V24	Ports 80-88		280-CTC		16
			280-DART		35
HRG1B von RB	Ports 00-7F		?	155	2,14
Schmidtke 802.Karte	B000-BFFF		?	245	19
	Ports D0, D1		?	138	15
EG64MBA	Port DF	bleibt unberücksichtigt	#1		
EG64MBA+ <6>	Ports DE, DF	"	"	"	"
256K-Banker <10>	Port EC	"	"	"	"
	(u. ED mit dieser Karte)	"	"	"	"
CP/M-Banker <6>	Ports 50-5F	"	"	"	"

#1 diese Baugruppen werden nicht gelesen und brauchen deshalb nicht in die Buffersteuerung miteinbezogen zu werden, das IN A,(ODFH) des EG64MBA dient nicht der Datenaquisition, es setzt nur das 74LS259-Latch zurück, ein Datenwort wird nicht ausgegeben

Im Normalfall liegen alle memory-mapped (mm) Baugruppen CPU-seitig des Treibers, so daß dann einzig MERQ* (IC16, 74LS367, Pin5) an die Schaltung geführt werden muß. Wenn aber auch nur eine mm Baugruppe auf dem ECB-Bus steckt, müssen alle anderen mm Freigabesignale einzeln an die Schaltung geführt werden.

Durch Vorschalten weiterer AND-Gatter vor die Eingänge des 74LS30-ICs (z.B. 74LS08, 74LS11, 74LS21) kann die Anzahl der verwendbaren Eingänge für Freigabesignale beliebig erhöht werden (Abb.10).

Die DMA-Fähigkeit der ECB-Busbuffer alleine reicht nicht aus. Es muß durch Verbinden der Systembus-Leitung BUSAK* mit CCDBS/STADBS* und DODBS/ADDDBS* auch dafür gesorgt werden, daß beim DMA-Zugriff die Buffer der CPU-Signale in den hochohmigen Zustand versetzt werden. Das kann entweder auf der Eingangskarte durch Querverdrahtung der entsprechenden Pins von CN2 bzw. CN3 oder durch feste Verdrahtung auf dem CPU-Board gemäß Abb.11 erreicht werden.

Da für die Richtungssteuerung des Datenbuffers ohnehin einige ICs nötig waren, und der ECB-Bus auch IM2-fähig ist, ließ sich mit etwas Mehraufwand auch gleich eine Vektorisierung der Interrupts des 25ms-Timers und des Floppy-Controllers realisieren. Dafür ist Voraussetzung, daß auf dem ECB-Bus auf einer Karte ein freier Z80-CTC-Kanal (2) vorhanden ist, über den durch Software das gemeinsame INT*-Signal des Systems umgeleitet werden kann. Der CTC-Kanal ist dann im Zähler-Modus mit der Zählkonstanten 1 und Triggern durch eine negative Flanke zu programmieren.

Die direkte Verbindung des INT*-Signals (im Schaltplan INTEXP*) von der Floppy-Baugruppe zur CPU ist dafür aufzutrennen. Das Signal INTEXP* ist stattdessen an den entsprechend bezeichneten Punkt der Schaltung und an den freien Triggereingang des CTC-Kanals zu legen. Die INT*-Leitung des ECB-Bus ist an den Punkt INTECB* zu legen und der Ausgang INTCPU* ist mit dem INT-Eingang der CPU (Z80, Pin16) zu verbinden.

Damit ist gewährleistet, daß nach dem Einschalten oder Drücken des RESET-Knopfes immer der Interrupt vom FDC-Board direkt zur CPU durchgeschaltet wird und das Betriebssystem im IM1 arbeiten kann. Wenn allerdings andere Baugruppen auf dem ECB-Bus in Echtzeit im IM2 betrieben werden sollen, müssen auch die systemeigenen Interrupts dieses Spiel mitmachen können und das geht nur dann, wenn ihnen der CTC-Kanal beim INT-Acknowledge einen Vektor zur Verfügung stellt. Die Schaltung leistet es, daß ein LD A,xxxx xx1xB; OUT (0EDH),A das INT* Signal des FDC-Boards über den CTC umleitet.

Die Software muß dann aber Rücksicht auf diesen Umstand nehmen. In der INT-Vektor-Tabelle darf nicht einfach 4012H stehen (INT-Vektor des DOS), da die Serviceroutine des DOS nicht mit RETI sondern mit RET endet und der Z80-CTC dann nicht weiß, wann er seinen IEO-Ausgang wieder auf High zurücknehmen soll. Es muß folgender Umweg eingeschlagen werden:

```
INTAB  DEFW  SERV1    ;Anfang INT-Tabelle
        DEFW  SERV2    ;nächster Vektor
        DEFW  .....    ;u.s.w.
        ....
        DEFW  CTCN     ;Vektor für Serviceroutine des INT vom CTC-Kanal
        ....          ;N, über den die System-Interrupts laufen
        DEFW  LETZT    ;Ende der Vektor-Tabelle
        ....
CTCN   CALL  4012H    ;Serviceroutine des DOS
        EI
        RETI
```

Außerdem bleibt es dem Anwender offen, während des IM2-Betriebes anderer Baugruppen den 25ms INT verhungern zu lassen. Wenn beim Label CTCN einfach ein EI, RETI steht, wird 37E0H nicht mehr gelesen und daraufhin erfolgt kein weiterer Interrupt des Timers mehr. Eine Reinitialisierung des Timer-INTs geschieht durch Umprogrammieren des Vektors auf ein anderes Label, das wieder den CALL 4012H enthält, und einmaliges Lesen von 37E0H.

Durch Zurücksetzen von D1 in Port EDH wird wieder der INT des FDC-Boards direkt an die CPU durchgeschaltet und alle Interrupts des ECB-Bus werden unterbunden.

Mit der für die Interrupt-Steuerung benötigten Portdecodierung und mit dem Latch wurden noch einige zusätzliche Funktionen realisiert. Zunächst wurden auch die Portfreigabesignale OUTECH*, OUTEEH* und OUTEFH* hergeleitet. Davon kann OUTECH* zur Freigabe des 74LS273 Latch auf dem 256K-Banker <10> benutzt werden. Dieses Signal wird direkt an den Pin11 des 74LS273 geführt, wobei dann aber die bisher dahin führende Leitung zu durchtrennen ist. Damit ist dann das 74LS30 IC auf dem Banker überflüssig und auch die Leitungen A0 bis A5 sowie IORQ* und WR* brauchen nicht mehr dorthin geführt zu werden (A6 und A7 werden dort nicht nur zur Portdecodierung sondern auch noch zur Erzeugung der 8Bit-Refreshadresse benötigt und müssen deshalb weiterhin angeschlossen bleiben).

Mit Bit 0 in Port EDH wurde ein softgesteuertes Invertieren des zum 256K-Banker führenden Signals A15 realisiert. Damit läßt sich mit D0=0 an Port EDH (Einschaltzustand) vorgeben, daß die unteren 32K Adreßbereich des 280 gebankt werden können, während D0=1 an Port EDH das Banking der oberen 32K festlegt.

Und schließlich kann mit D2=1 an Port EDH ein gemeinsames Invertieren der Adressen A14 und A15 erreicht werden. Die dabei gewonnenen Signale A14' und A15' können anstelle der Signale A14 und A15 an die Decoderschaltung für die 16K (ROMs und mm I/O), IC25, 74LS139, Pins 14 und 13 geführt werden und gestatten dadurch ein Verlegen dieser Komponenten an das obere Speicherende 48-64K. In diesem Bereich funktionieren dann zwar die ROMs nicht mehr, wenn man aber dann noch ein 74LS32 IC spendiert und mit dem Signal ED,3 (D3 von Port EDH) ein programmierbares Abschalten der ROMs einbaut (Abb.10), hat man auch noch die Funktionen des Omikron Mappers (kompatibler Selbstbau in <9>) für den Betrieb von CP/M realisiert. Dieser leistet (wenn auch mit einem anderen OUT-Befehl) ebenfalls das Einstellen der Systemkonfiguration:

```
0000H-F7DFH  RAM <7> #1
F7E0H-F7FFH  Floppy, Drucker
F800H-FBFFH  Tastatur <7>
FC00H-FFFFH  Video-RAM
#1 der Omikron Mapper für den TRS 80 leistet selbst eine vollständige
  Decodierung des Bereichs des Sonder-ROMs des GENIE
```

Ein Patchen der CP/M Systemdiskette ist sehr einfach. Auf den Systemspuren läßt sich relativ einfach die Sprungleiste des Loader Bios finden, die aus einer Reihe von Jumps besteht (C9 nn nn C9 nn ...). Daran schließen sich die Disk Parameter Tabellen mit sehr vielen 00H an und unmittelbar danach stehen die Befehle LD A,40H; OUT (50H),A. Diese beiden Befehle sind durch die Befehle LD A,xxxx 11xxB; OUT (0EDH),A zu ersetzen. Damit läuft das CP/M des Omikron Mappers dann auch ohne denselben.

Wenn diese Eingriffe auf dem CPU-Board gemacht werden, ist der Computer aber von der ECB-Bus-Eingangskarte abhängig, weil von dieser dann lebenswichtige

Signale geliefert werden. Der ECB-Bus braucht selbst nicht angeschlossen zu sein.

Von dieser Abhängigkeit kann man sich durch einen Mehrfach- (oder mehrere Wenigfach-) Umschalter lösen. Abb.9 zeigt, wie man damit entweder die ursprünglichen Signale des CPU-Boards oder die steuerbaren Signale dieser Schaltung an die entsprechenden Stellen des CPU-Boards führt, und damit das Gerät entweder wie bislang ohne ECB-Eingangskarte oder mit derselben und allen ihren Mehrleistungen laufen lassen kann.

Um den Umfang der (zusätzlich zu denen des Systembus) an die Eingangskarte geführten Leitungen etwas zu reduzieren, bietet es sich an, IC7, 74LS30 auf der Eingangskarte nicht zu bestücken und dieses IC im Computer an geeigneter Stelle huckepack mit den Pins 7 und 14 auf ein anderes Dill4 IC zu löten und die internen Freigabesignale fest dorthin zu verlegen (Pins 1-6, 11-12). Dann braucht nur noch das Signal INTERN von Pin8 des 74LS30 an die Eingangskarte (CN2, Pin25c) geführt zu werden.

Wenn die zusätzlichen Features der Eingangskarte voll genutzt werden sollen, können alle benötigten Signale (SYSRES* A15* A14' A15' ED,3 DUTECH*) über freie Verdrahtung an unbelegte Pins von CN2 gelötet werden. Im Layout sind nur die Signale INTCPU* INTEXP* INTECB* und INTERN über CN2 geführt. Dadurch können alle Verbindungen mit dem Computer (nicht nur der Systembus) über CN2 laufen.

Wer diese Variante des ECB-Bus bauen möchte, kann bei mir für 10 DM (incl. Rückporto) eine geätzte aber ungebohrte Karte im Europaformat bekommen. Sollte irgendjemand die Karte am Komtek 1 betreiben wollen, kann ich entsprechende Informationen zur Verfügung stellen, die eine Nutzung auch dort ermöglichen. Der Anschluß an den TRS 80 ist nicht ohne Sondermaßnahmen möglich, weil die internen Datentreiber IM2-Betrieb verhindern. Ein Hardware-Patch, der dagegen hilft, ist in <8> beschrieben. Eine Herleitung der Signale IORQ* MERQ* RD* und WR* aus den daraus gewonnenen Signalen IN* OUT* MWR* und MRD* des TRS 80 Systembus mit Gattern erzeugt Signale mit nicht ganz richtigem Timing, die eventuell Schwierigkeiten bereiten können; hier ist das Abgreifen der originalen Signale eine bessere Lösung.

Aufbau und Anschließen der Karte

Entgegen der üblichen Technik, solche Karten doppelseitig geätzt zu entwerfen, wurde hier ein einseitiges Layout mit einer unerschämten Menge Drahtbrücken bevorzugt, was das Durchkontaktieren an IC-Sockel-Pins vermeidet. Damit wird beim Aufbau aber viel Fummelei und beim Testen viel Ärger erspart.

Der Stecker CN1 (VG64, a,c, Stifte, gewinkelt) wird auf den ECB-Bus gesteckt. Für den Anschluß an den Systembus wurden 2 Varianten vorgesehen. Durch Einbau einer VG64 a,c-Buchsenleiste bei CN2 von der Lötseite (Abb.8) muß durch einen entsprechenden Drahtverhau zum Gegenstück für den 50poligen Platinenrandstecker des CPU-Boards die Anordnung der Signale umsortiert werden. Wer allerdings ein ästhetisches 50poliges Flachkabel bevorzugt, kann CN2 unbestückt lassen und stattdessen in die nicht angeschlossenen Löttaugen von CN3 von der Lötseite einen geeigneten 50poligen Stecker einlöten. Dann wird aber der zum Umsortieren der Signale nötige Drahtverhau nur auf die Karte zwischen CN2 und CN3 verlagert.

Außer den reinen Bus-Signalen des Computers müssen bei Bedarf auch noch die aktiven Steuersignale der Eingangskarte an den Computer zurückgeführt werden und die im Computer abzugreifenden Freigabesignale an die Eingangskarte geführt werden. Dafür sind am oberen Platinenrand entsprechende Lötunkte vor-

handen. Die wichtigeren Signale sind aber auch an Lötunkte von CN2 geführt, womit dann alle Signale über CN2 (und bei entsprechender Querverdrahtung auch CN3) mit dem Computer verbunden werden können. Dafür reicht dann aber der 50polige Platinenrandstecker des CPU-Boards mit seinen 3 freien Pins nicht mehr aus. Da muß dann noch eine zusätzliche fliegende Steckverbindung herhalten, die diese zusätzlichen Signale übergibt.

Seckerbelegungen

CN1 (ECB-Bus)			CN2 (vom Systembus)		
Reihe a	NR.	Reihe c	Reihe a	Nr.	Reihe c
GND	32	GND	GND	32	GND
BUSAK*	31	RESET*	BUSAK*	31	RESET*
A9	30	MERG*	CLOCK	30	nc
A13	29	CLOCK	MERG*	29	RFSH*
RFSH*	28	A15	IORQ*	28	RD*
IORQ*	27	A12	M1*	27	WR*
#1	26	#1	HALT*	26	nc
#1	25	HALT*	nc	25	INTERN
#1	24	RD*	nc	24	INTEXP*
#1	23	#1	NMI*	23	INTECB*
-12V	#2	WR*	A11	22	A8
#1	21	INT*	A10	21	A14
M1*	20	NMI	A13	20	A9
#1	19	#1	A12	19	A15
A14	18	A10	nc	18	nc
#1	17	A11	nc	17	nc
#1	16	IE0	nc	16	nc
-5V	15	#1	nc	15	nc
#1	14	D1	A7	14	A6
+12V	13	#1	A5	13	A4
#1	12	A14	A3	12	A2
BUSRQ*	11	IE1	A1	11	A0
Wait*	10	#1	BUSRQ*	10	nc
A6	9	A7	nc	9	WAIT*
A5	8	A8	nc	8	nc
A4	7	A1	D5	7	D0
A2	6	A3	D6	6	D7
D4	5	A0	D3	5	D2
D3	4	D2	D4	4	D1
D6	3	D7	nc	3	INTCPU*
D5	2	D0	nc	2	nc
+5V	1	+5V	(+5V)	1	(+5V)

Stückliste

ICs:

4 x 74LS245
 1 x 74LS174
 1 x 74LS139
 1 x 74LS125
 1 x 74LS86
 1 x 74LS32
 1 x 74LS30
 1 x 74LS11
 1 x 74LS08
 1 x 74LS00

Widerstände:

2 x 10k

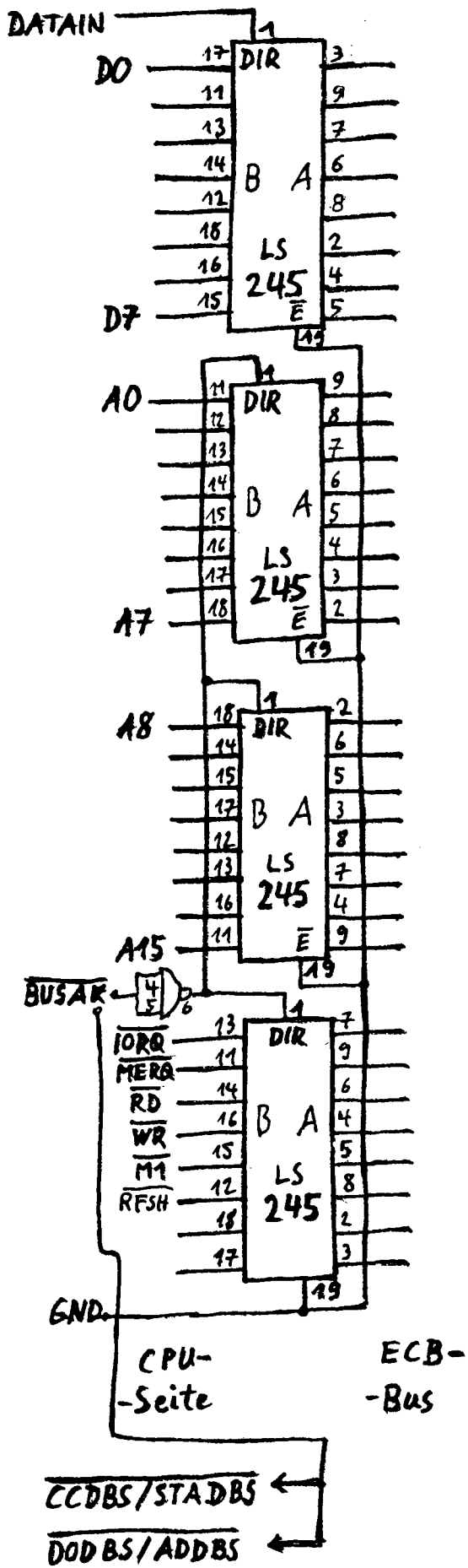
Sonstiges:

1 x VG64 a,c Stifte
 gewinkelt
 1 x VG64 a,c Buchsen
 gerade
 1 x Leiterplatte
 jede Menge Draht
 viel Geduld

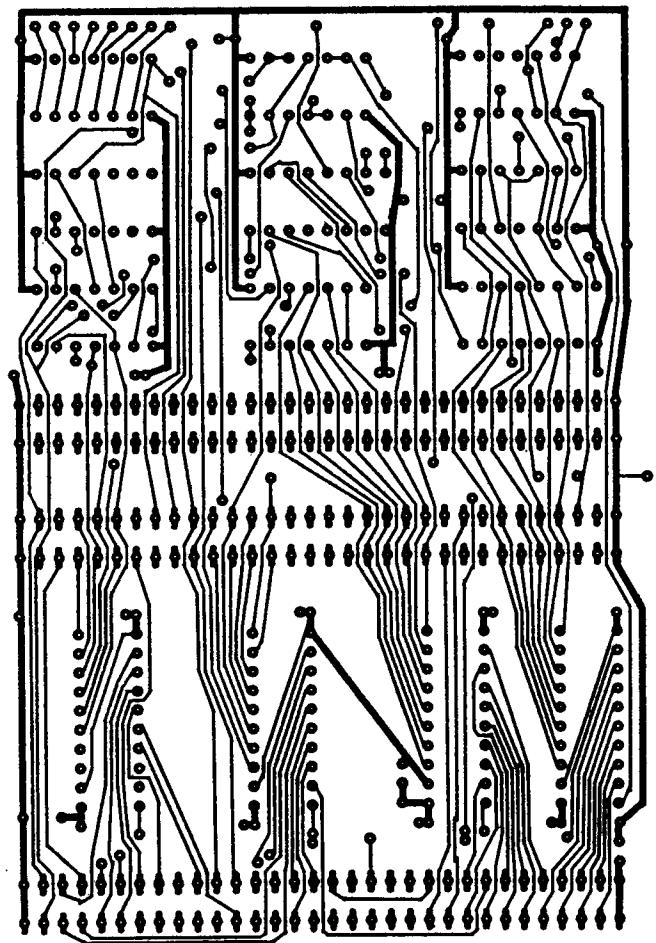
 #1 diese Pins sind zum Teil nicht
 belegt oder führen Signale, die
 hier nicht benutzt werden
 #2 die Lage von -12V wird nicht
 einheitlich gehandhabt, diese
 Spannung kann auch woanders
 liegen (muß bei jeder gekauften
 Karte geprüft werden)

Literatur

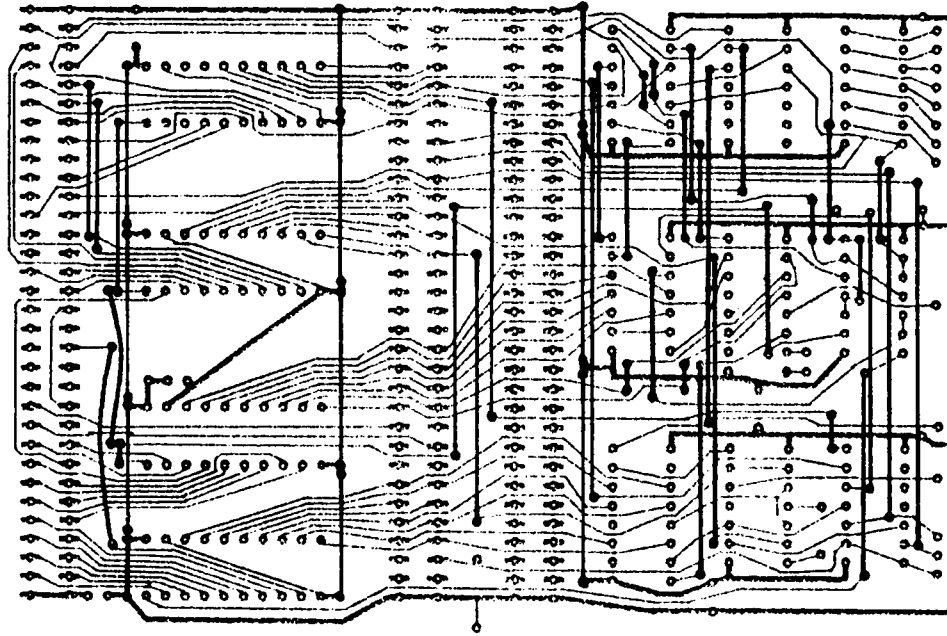
- <1> Verbindend - Der ECB-Bus / Johannes Assenbaum / c't 12/85, 60-61
- <2> Z80 CTC - Einsatz und Programmierung / Rolf Keller / c't 4/85, 92-94
- <3> Interrupt-Priority-Encoder und Vektorgenerator für IM2 mit dem Z80 CTC / H. Bernhardt / Info des GENIE/TRS 80- User-Club Bremerhaven 12/85, 10-11
- <4> Direkter Speicherzugriff; Schneller als die CPU - So arbeiten DMA-Controller-ICs / Rolf Keller / c't 8/85, 80-83



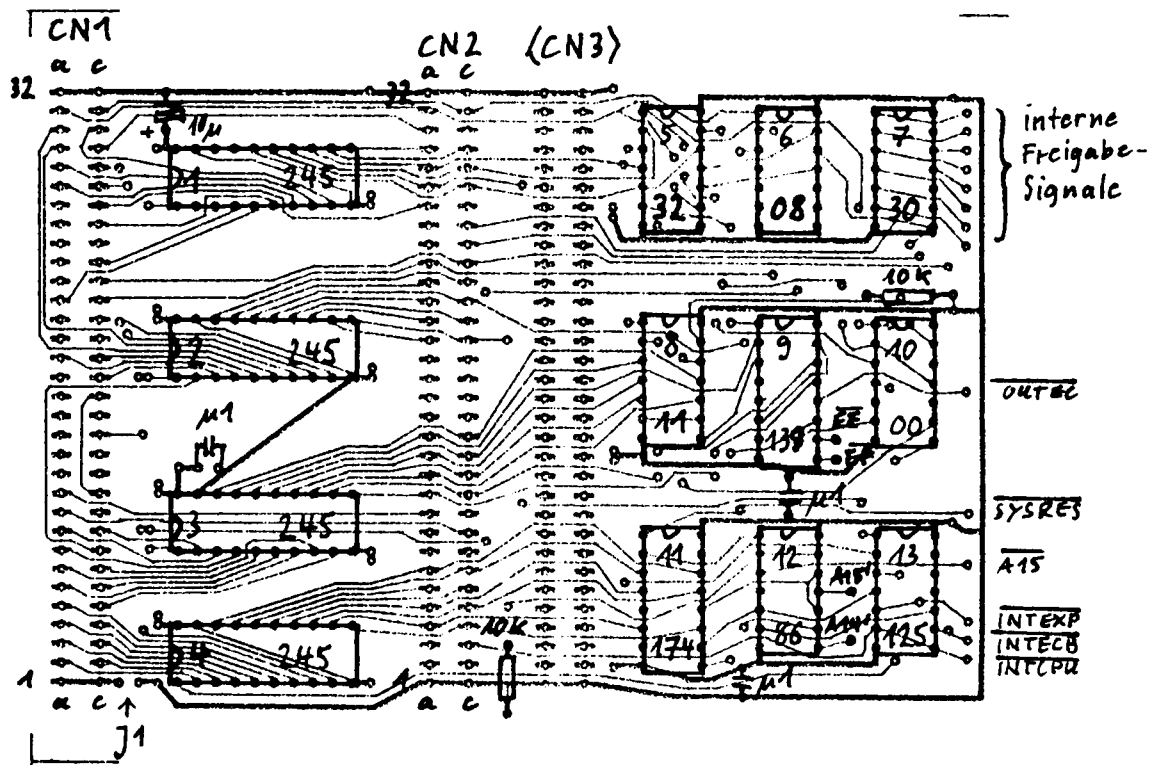
2) Pufferung des ECB-Bus



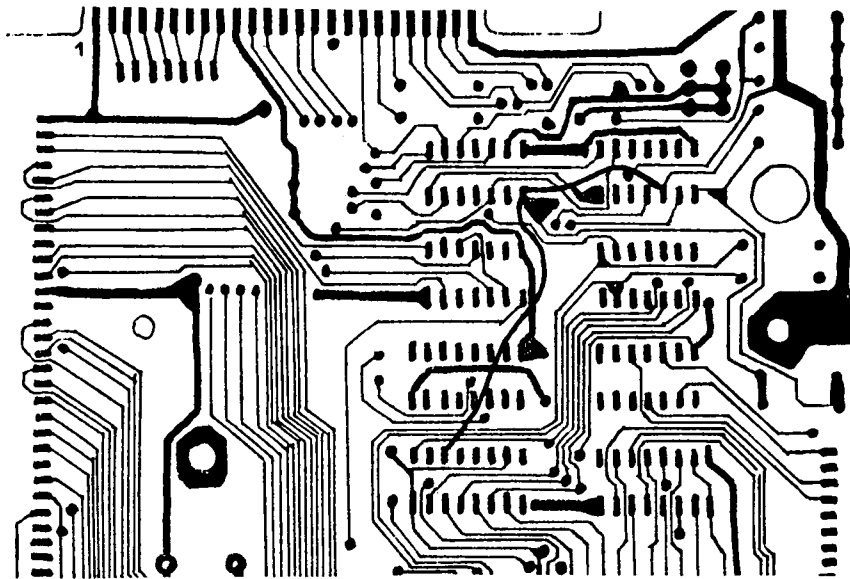
3) Layout der Eingangskarte



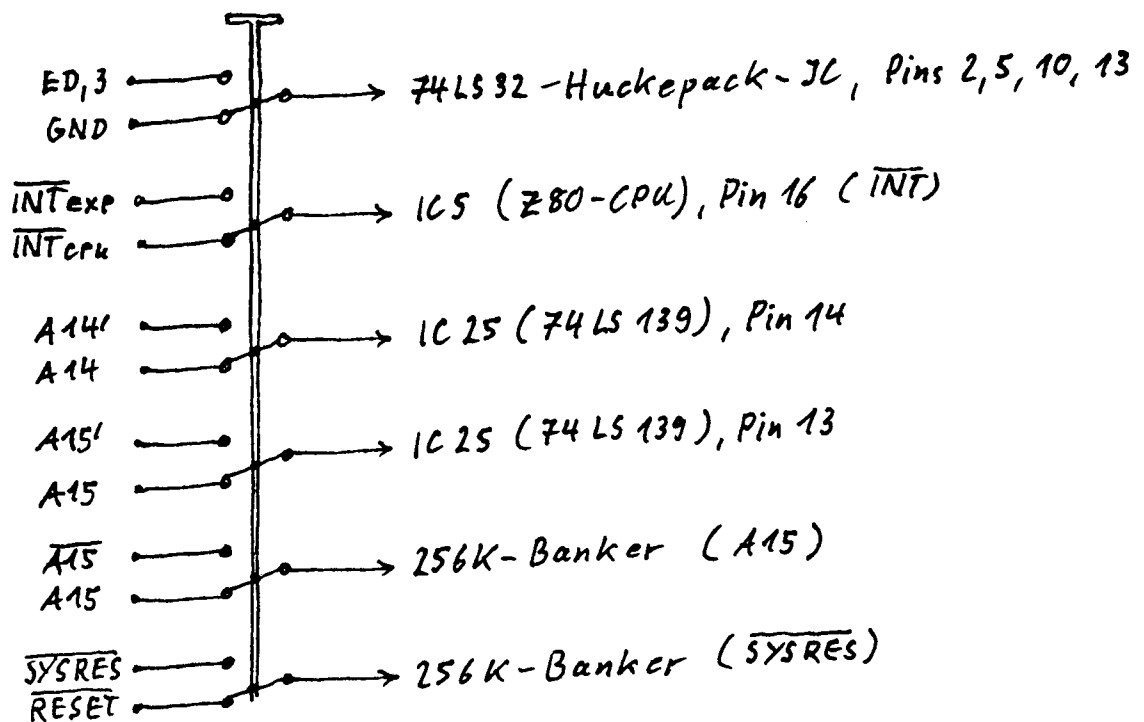
4) Drahtbrückenplan, die Drahtbrücken sind vor der Bestückung aller anderen Bauteile zu legen



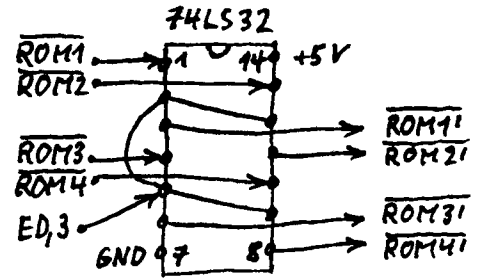
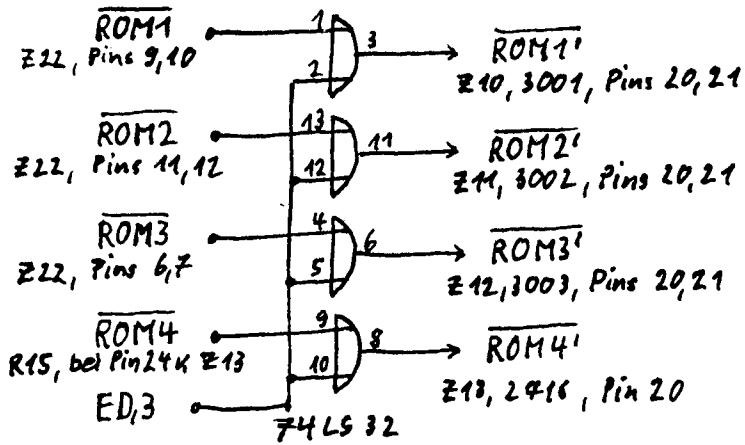
5) Bestückungsplan, J1 ist nur dann zu legen, wenn im GENIE ein stärkeres Netzteil eingebaut wurde, das den ECB-Bus mitversorgen soll



8) Verbinden von BUSAK* mit CCDBS/STADBS* und DODBS/ADDBS*



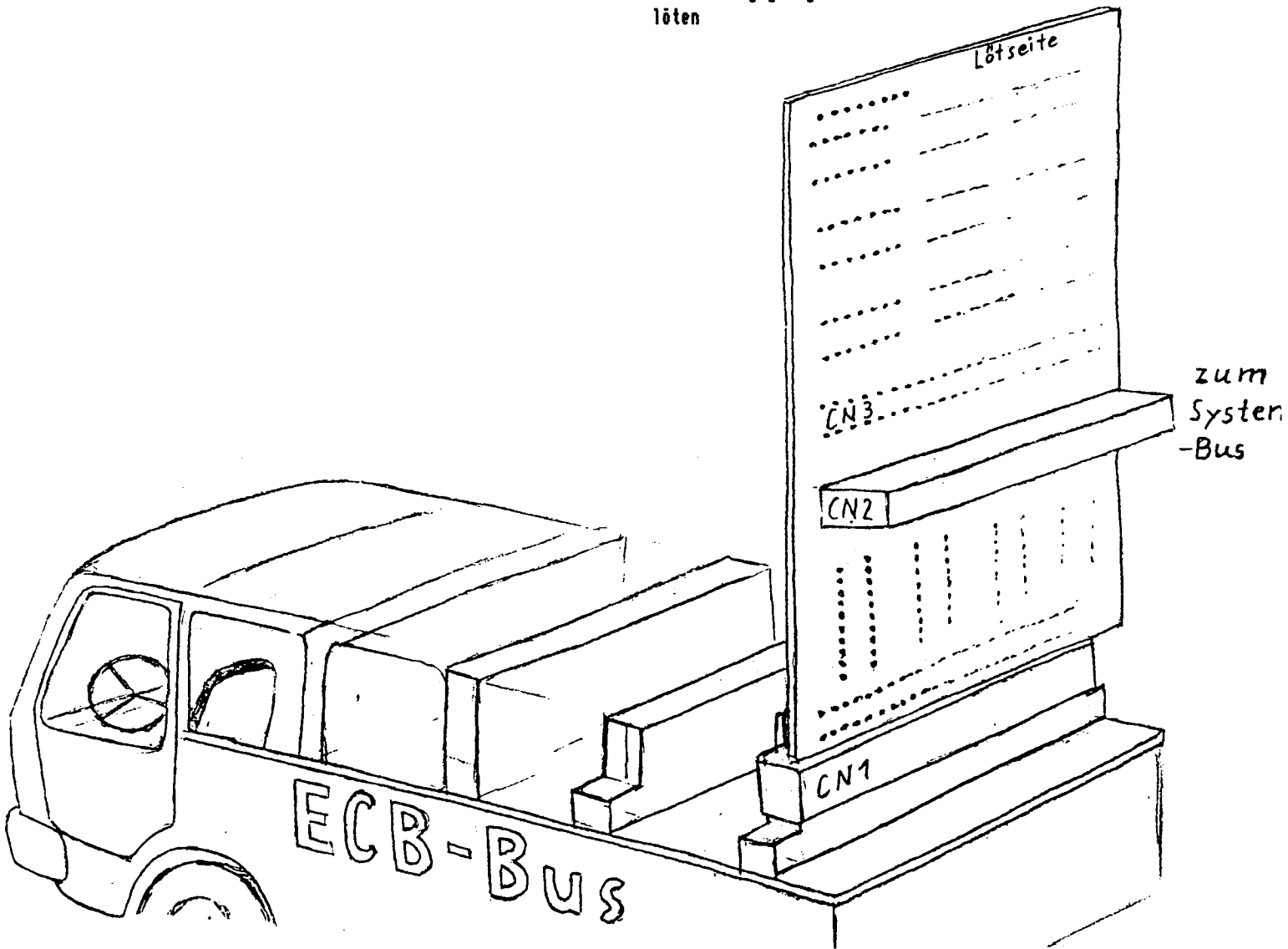
9) Umschalten zwischen originalen GENIE-Signalen und steuerbaren Signalen der Eingangskarte



6) Beschaltung des Huckepack 74LS32 auf dem CPU-Board zum Ausblenden der ROMs

ROM1*	Z22, 74LS156, Pins 9,10	ROM1'*	Z10, 3001, Pins 20,21
ROM2*	" " " 11,12	ROM2'*	Z11, 3002, Pins 20,21
ROM3*	" " " 6,7	ROM3'*	Z12, 3003, Pins 20,21
ROM4*	R15, Seite zu Pin 24 von Z13 (ROM4)	ROM4'*	Z13, 2716, Pin 20

diese Signale sind an die aus der Fassung gebogenen Pins zu löten



7) Verbinden des ECB-Bus mit dem Systembus des Computers über die Eingangskarte

----- ECB-BUS Belegung (CONITEC-Norm) -----

Reihe	a	b	c	Bedeutung:
1:	+5V	xxx	+5V	: +/-5V/12V: Betriebsspannungen
2:	D5	xxx	D0	: UBAT: Akku (+3.6 Volt)
3:	D6	xxx	D7	: GND: gemeinsame Masse
4:	D3	xxx	D2	:
5:	D4	xxx	A0	: D0-D7: 8-Bit-Datenbus
6:	A2	xxx	A3	: A0-A15: 16-Bit-Adressbus
7:	A4	xxx	A1	: A16-A19: 20-Bit-Adressen
8:	A5	xxx	A8	: A19/MS: MASTER/SLAVE-Signal
9:	A6	xxx	A7	:
10:	/WAIT	xxx	xxx	: /RD: Lesen
11:	/BUSRQ	xxx	...IEI	: /WR: Schreiben
12:	A18 !	xxx	: A19/MS !	: /MREQ: Speicherzugriff
13:	+12V	xxx	: xxx	: /IORQ: Ein/Ausgabe
14:	-12V !	xxx	: D1	:
15:	-5V	xxx	: -	: /NMI: Vorrang-Interrupt
16:	-	xxx	:...IEO	: /INT: normaler Interrupt
17:	A17 !	xxx	A11	: IEI: INT-Ketteneingang
18:	A14	xxx	A10	: IEO: INT-Kettenausgang
19:	-	xxx	A16 !	:
20:	/M1	xxx	/NMI	: /BUSRQ: DMA-Busanforderung
21:	xxx	xxx	/INT	: /BUSAK: DMA-Busfreigabe
22:	-	xxx	/WR	: /BAI: DMA-Ketteneingang
23:	.../BAI !	xxx	-	: /BAO: DMA-Kettenausgang
24:	: UBAT	xxx	/RD	: /RDY: DMA-Handshake
25:	:.../BAO !	xxx	/HALT	:
26:	/RDY !	xxx	/PCL	: /RESET: Rücksetz-Eingang
27:	/IORQ	xxx	A12	: /PCL: Initialisieren
28:	/RFSH	xxx	A15	: /WAIT: CPU-Wartezyklus
29:	A13	xxx	CLOCK	: /HALT: CPU gestoppt
30:	A9	xxx	/MREQ	: /M1: 1. Maschinenzyklus
31:	/BUSAK	xxx	/RESET	: /RFSH: Auffrischzyklus
32:	GND	GND	GND	: CLOCK: CPU-Takt (4-8 MHz)

xxx = reserviert für Erweiterungen ... = Kettenleitungen (Daisy Chain)
 - = zur freien Verwendung ! = abweichend von der KONTRON-Norm

----- ECB-Busnormen anderer Hersteller auf einen Blick -----

	CONITEC	E-LAB	ELZET	J&K	KONTRON	MARFLOW	miro	otr
c10	xxx	A21	--	A16	MBS0	PHI/2	--	A16
a12	A18	A18	A18	/BAI	/BAI	A18	A18	/BAI
c12	A19	A19	A19	A17	MBS1	A19	A19	A17
c13	xxx	-12V	--	A18	MBS2	-12V	A20	A18
a14	-12V	/MMUDIS	-12V	A19	MBS3	xxx	--	A19
a16	--	2PHI	2PHI	2PHI	2PHI	nPHI	2PHI	--
a17	A17	A17	A17	/BAO	/BAO	A17	A17	/BAO
c19	A16	A16	A16	--	MBS5	A16	A16	--
a21	xxx	BDCL	--	--	MBS6	xxx	A22	--
a22	--	/RDY	/BOOT	/RDY	MBS7	xxx	A23	--
a23	/BAI	/BAI	/BAI	/PF	DPR	/BAI	/BAI	--
c23	--	A20	/VBLANK	/SEL	MBS4	xxx	A21	--
a25	/BAO	/BAO	/BAO	--	nPHI	/BAO	/BAO	--
a26	/RDY	MEMD	--	/DESLCT	WREN	xxx	/RDY	/DESLCT

Obige Tabelle ohne Gewähr. Achtung! Die unterschiedliche Belegung der -12V-Leitungen kann bei einigen Kombinationen von Karten und Bussen verschiedener Hersteller zu starker Rauchentwicklung bei der Inbetriebnahme führen! Deshalb -12V vorsichtshalber immer vorher abklemmen.

