

Rolf-Dieter Klein

Der 32-Bit-68020-Selbstbaucomputer

NDR-Klein-Computer ganz groß

Das NDR-Klein-Computer-Konzept trägt mühelos bis zum echten 32-Bit-Computer. Damit werden Rechenleistungen erreichbar, wie sie vor kurzem nur bei Minicomputern üblich waren (Gleitkomma-Addition, Multiplikation in 3 μ s). Ein schneller Gleitkomma-Prozessor steigert die Geschwindigkeit beim numerischen Rechnen enorm. Der 68020 verfügt außerdem über einen 256 Byte (64 Langworte) großen Instruktions-Cache-Speicher, der alle Befehle mitspeichert. Bei erneutem Zugriff auf eine dort bereits abgespeicherte Adresse benutzt er den Inhalt des Cache. Damit werden zum Beispiel viele Schleifen sehr schnell und können oft ganz ohne äußeren Zugriff arbeiten.

Viele Leser kennen den NDR-Klein-Computer noch mit dem 8-Bit-Datenbus. Es ist relativ einfach möglich, diesen Datenbus auf 16 oder 32 Bit zu erweitern. Bei einer Erweiterung auf 16 Bit wird der Bus in der Mitte zweigeteilt. Von der CPU aus gesehen links wird der Datenbus mit den Leitungen D15 bis D8, rechts mit den Leitungen D0 bis D7 angesteckt. Dies hat den Vorteil, daß man alle 8-Bit-Karten aus unserem System

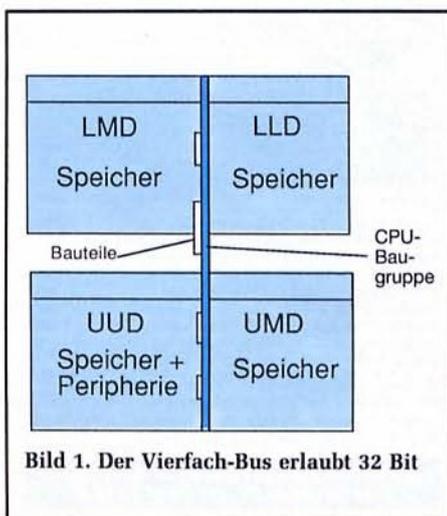
ungeändert weiterverwenden kann. Nur den Speicher muß man jetzt immer in doppelter Ausführung bereithalten. Bei einer Erweiterung muß man auf der linken und rechten Bus-Platine je eine neue Speicherkarte einsetzen. Beim 32-Bit-Bus wird jetzt auch in die Breite gegangen (Bild 1). Diesmal benötigt man noch zwei weitere Bus-Baugruppen dazu, der Datenbus wird nun viergeteilt. Die CPU sitzt in der Mitte des Systems. Dabei sind links neben der CPU etwas mehr Steckplätze als rechts vorhanden, denn die Peripherie, die vom Grundprogramm unterstützt wird, wird an den Datenbus D31 bis D24 angeschlossen. Da die Peripherie manchmal Stecker an der Schmalseite der Platten besitzt, ist dies der günstigste Platz. Das Grundprogramm, enthält alle wichtigen Routinen, die man zum Start des Systems benötigt. Neben einem Editor auch einen Assembler, der um die neuen Befehle für den 68020 (CPU) und den 68881 (FPU) erweitert wurde. Ferner sind im Grundprogramm Routinen für den Floppy-Betrieb eingebaut (siehe auch Sonderheft „Mikrocomputer Schritt für Schritt“). Die normalen NDR-Klein-Computer-Baugruppen für acht Bit besitzen eine

einreihige Stiftleiste zur Verbindung mit dem Bus. Die CPU-68020 besitzt zwei zweireihige Stiftleisten. Auf der Busbaugruppe gibt es zwei Steckplätze zur Auswahl, in die man die CPU stecken kann. Bild 2 zeigt die Belegung der Busanschlüsse. Auf der Unterseite der Busbaugruppen muß man die Leitungen D0 bis D7, sowie IORQ, MREQ, RD und WR auftrennen. Jede Bushälfte muß getrennt versorgt werden. Achtung: Man trenne auch RD und WR, obwohl sie beim 68020 auf der Platte verbunden sind, um mit anderen 16- und 32-Bit Karten kompatibel zu bleiben. Alle anderen Signale müssen an beiden Bushälften anliegen.

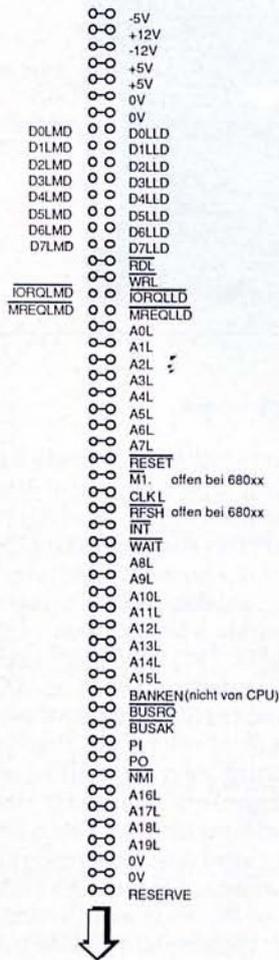
Beim 68020 gibt es vier Datenbusteile. Das Leitungsbündel D31 bis D24 wird als UUD (Upper Upper Datenbus) bezeichnet. Bei einem Speicherzugriff werden hier alle geraden Adressen angesprochen (also z. B. 0,4,8,...). D23 bis D16 gehören zum UMD, also Upper Middle Datenbus, D15 bis D8 zum LMD, also Lower Middle Datenbus, und D7 bis D0 bilden den LLD (Lower Lower Datenbus).

Bild 3 zeigt eine Zusammenstellung und Anordnung von Baugruppen, wie sie für den Betrieb des CP/M-68k-Systems nötig sind. Alle Speicherbaugruppen müssen in vierfacher Ausfertigung vorhanden sein. Man benötigt mindestens vier ROA-64-Baugruppen, auf denen das Betriebsprogramm sowie ein kleiner lokaler Speicher untergebracht sind. Wenn man mit CP/M arbeiten will, so braucht man weiteren Speicher, hier vier DYN-256-Karten, die einen Speicher von 1 MByte bereit stellen. CP/M würde auch mit 128 KByte auskommen, jedoch sind die 256-KByte-Chips nicht mehr teuer und vier Baugruppen benötigt man in jedem Fall. Zum Beispiel wird vom neuen BIOS (siehe mc 1986, Heft 4) eine RAM-Floppy unterstützt, die sehr viel Geschwindigkeit bringt.

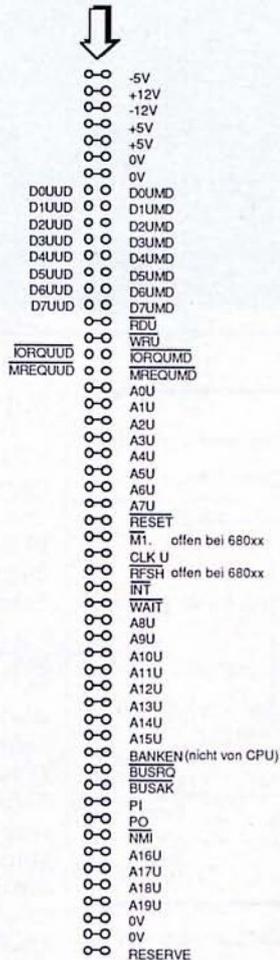
Das neue Grundprogramm (Version 5.x) benötigt insgesamt acht 2764-EPROMs. Es wird auf die ROA-Baugruppen gesetzt. Man benötigt dabei vier RAM-Bausteine des Typs HM 6264, die als lokaler Speicher (z. B. für den Bildwiederhol-speicher) genutzt werden. Eine weiteres EPROM sitzt auf der 68020-Baugruppe selbst. Es sucht automatisch nach dem Grundprogramm. Nach einem RESET wird es eingeblendet. Damit sind vier Bank-Boot-Baugruppen eingespart. Für CP/M benötigt man ferner eine FLO2-Baugruppe (Floppy-Controller), eine GDP-Baugruppe (Grafik-Ausgabe) und eine KEY-Baugruppe (Tastatur).



Busansicht obere Hälfte



Busansicht untere Hälfte



◀ Bild 2. Die Signale am Bus

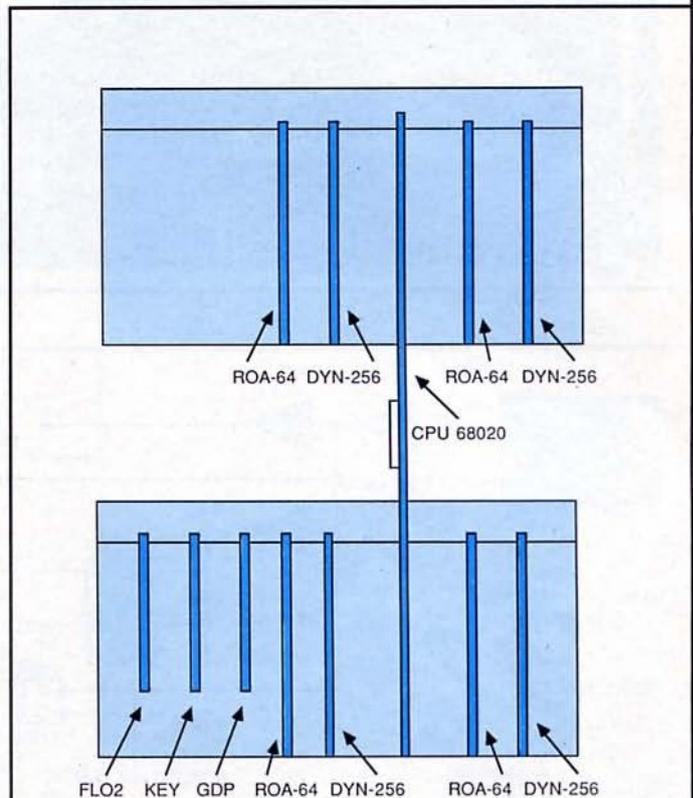


Bild 3. Diese Baugruppen werden für CP/M-68k benötigt

Die Hardware

Die CPU 68020 ist auf einer vierfach-Multilayer-Platine aufgebaut, weshalb hier im Heft auf den Abdruck des Layouts verzichtet wurde. Wenn man die Schaltung selbst aufbauen will, sollte man die Fädertechnik verwenden und gut auf Massen achten. Der Selbstbau ohne Leiterplatte ist allerdings nur etwas für Profis. Bild 4 zeigt das Foto der aufgebauten CPU-Karte. Bild 5 zeigt die Schaltung der Takt-Logik. Ein Quarzoszillator liefert der CPU einen Takt von 12 MHz. Alle Schaltungsteile wurden auch bereits mit 16 MHz für schnelle 68020-Chips getestet. Der Ausgang des Quarz-Oszillators wird über Nicht-Glieder gepuffert. Die CPU und die FPU (Floating Point Unit) werden über je einen 47-Ohm-Widerstand mit dem Takt verbunden. Der Bus wird über eigene Gatter versorgt. Über den Jumper J2 kann einmal der negierte Takt an den Bus geführt werden und einmal der nicht-negierte, aber auch

ein nicht-negierter Takt, der bei einem DMA-Zugriff in den Tri-State-Zustand gesetzt werden kann. Dazu dient der Speicher 74LS373. Die Tri-State-Ausgänge werden vom BGACK-Signal gesteuert, das auch andere 74LS373-ICs steuert.

Der Takteingang ist mit einem Widerstand nach +5 V verbunden. Wenn möglich sollte man aber die Brücke J2 offen lassen, denn alle neuen Baugruppen des NDR-Klein-Computers benötigen diesen Takt nicht. So kann man den Bus nämlich „sauber“ halten, denn 16 MHz sind kein Pappentier.

Die Reset-Logik

Bild 6 zeigt die Reset-Logik. Es gibt drei Reset-Quellen. Der Timer NE 555 liefert den Reset nach dem Strom-Einschalten. Hier wurde eine große Zeitkonstante gewählt. Das IC 74121 hat die Aufgabe, einen Schalter zu entprellen und einen kurzen Reset-Impuls zu erzeugen. Der

Reset-Impuls muß so kurz sein, daß durch den CPU-Stillstand keine Refresh-Zyklen (bei der DYN-256) ausfallen. Der Taster schließt einen Kondensator kurz, der sich über einen 10-kOhm-Widerstand langsam auflädt. Wenn der verwendete Taster stark prellt, so kann man anstelle des eingezeichneten 10-µF-Kondensators auch einen 47-µF-Kondensator verwenden.

Die beiden Reset-Signale von Timer 555 und Monoflop 74121 werden über Nicht-Glieder in einer Wired-Or-Schaltung zusammengeführt. Das Signal gelangt von dort aus direkt zum 68020 und zum 68881. Der 68020 kann per Befehl auch selbst einen Reset auslösen, der dann ebenfalls in Wired-Or-Verknüpfung an diesen Punkt führen kann. Der Widerstand von 150 Ohm ist für eine ältere XC68020-Maske gedacht. Zwei weitere Nicht-Glieder leiten das neue Reset-Signal auf den NDR-Klein-Computer-Bus. Alle Reset-Signale auf der CPU-Platine sind direkt mit dem Prozessor verbunden.

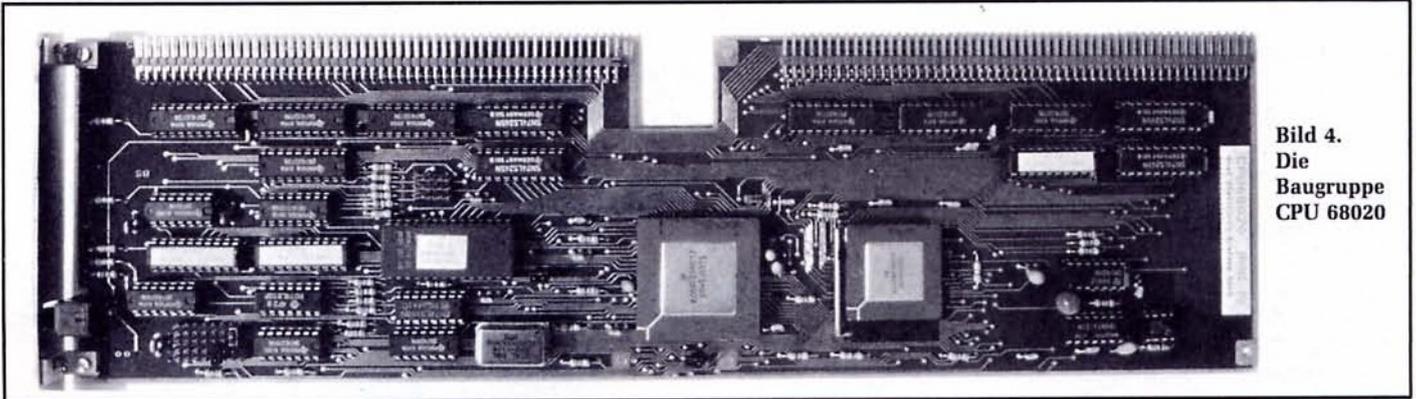


Bild 4.
Die Baugruppe CPU 68020

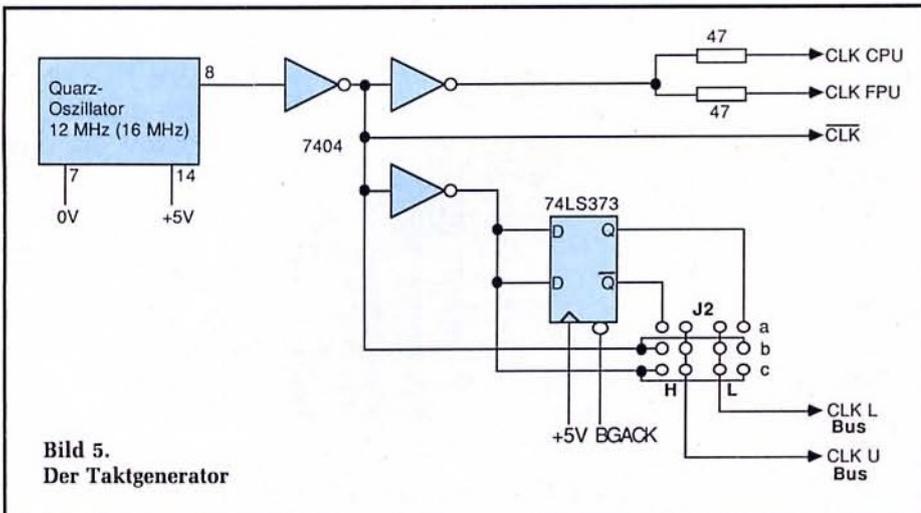


Bild 5.
Der Taktgenerator

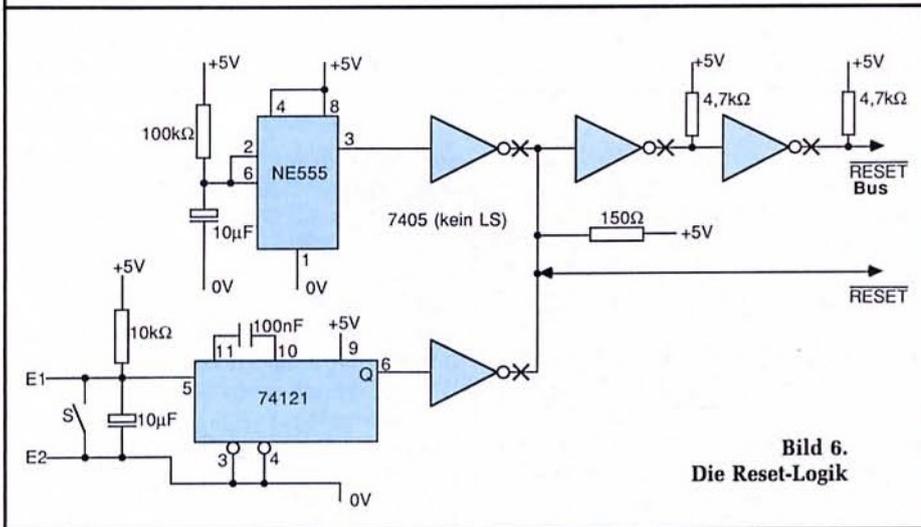


Bild 6.
Die Reset-Logik

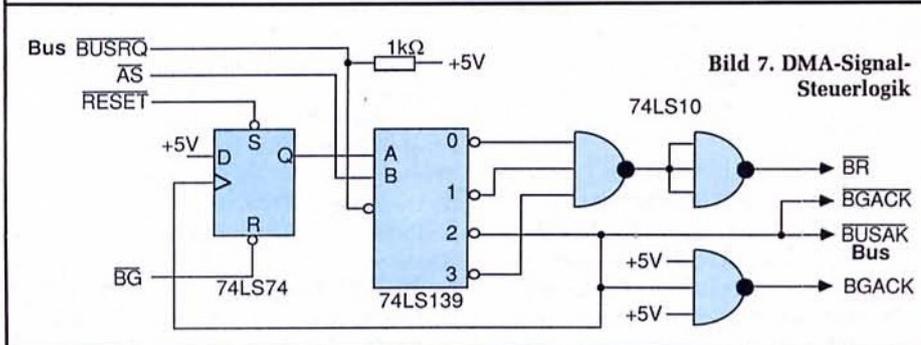


Bild 7. DMA-Signal-Steuerlogik

Mit DMA-Logik

Bild 7 zeigt eine Schaltung, die den DMA-Zugriff über die Signale BUSRQ und BUSAK verwaltet. Wird das Signal BUSRQ am Bus angelegt (aktiv Low), dann wird das Signal BUSAK von der Schaltung erst dann auf 0V gelegt, wenn der Bus wirklich frei ist. Wenn danach das Signal BUSRQ wieder auf High gelegt wird, so darf der 68020 wieder zugreifen und das Signal BUSAK wird dann ebenfalls wieder auf High gelegt. Die Schaltung sorgt dafür, daß das BUSRQ-Signal erst dann auf 0 geht, wenn alle Buszyklen des 68020 beendet sind. Dazu wird das eingehende Signal mit den internen Signalen AS (Adreß-Strobe) und BG (Bus Grant) verknüpft. Die Schaltung erzeugt zusätzlich das Signal BR (Bus Request) für den 68020. Nach Anlegen von BUSRQ wird zunächst BR auf 0 gesetzt, damit die CPU die Anforderung mitbekommt. Sie antwortet mit BG auf 0, was auch mitten in einem laufenden Bus-Zyklus geschehen kann. Daher wird durch den 74LS139 das Signal noch mit AS verknüpft und erst wenn AS auf 1 liegt, werden BUSAK und BGACK gegeben. BGACK wird invertiert und nicht-invertiert benötigt. Sind BUSAK und BGACK gesetzt, wird BR zurückgenommen (dafür erhält die CPU BGACK). Wenn das Signal BUSRQ wieder auf 1 gelegt wird, werden BUSRQ und BGACK zurückgenommen. Mit dieser Schaltung kann man den DMA-Zugriff völlig asynchron gestalten. Ein einfacher Test der Schaltung gelingt mit dem Signal HSYNC (von der GDP-Baugruppe), das man an den Eingang BUSRQ legt. Mit dem Oszilloskop kann man beobachten: Wenn BUSRQ auf 0V geht, so wird kurz danach der Bus freigegeben. Das System darf bei dieser „harten“ Probe nicht abstürzen. Wenn man VSYNC an die Leitung BUSRQ anlegt, so verschwinden ggf. der Cursor, da nun eine synchrone Abfrage des GDP nicht

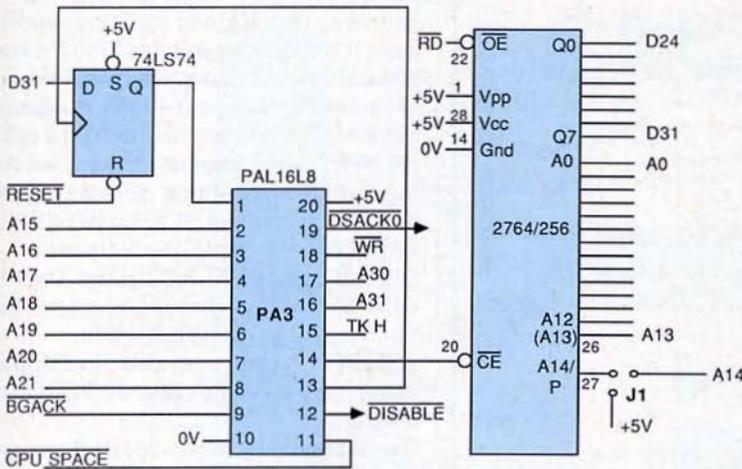


Bild 8. Die Boot-Logik

```

ENTER FILENAME (WITH EXTENSION) ---> PA3.PAL
PAL16L8

Pa 3 Pal fuer CPU 68020 NDR-Klein-Computer (C) 1986,Rolf-D.Klein

FFQAUS A15 A16 A17 A18 A19 A20 A21 NBSACK GND
NCPUSPAC NDISABLE NIOWRT NEPROMCE TKH A31 A30 NWR NDSACK0 VCC

IF (VCC) /NEPROMCE = /FFQAUS * /A15 * /A16 * /A17 * /A18 * /A19
* /A20 * /A21 * NCPUSPAC * /A30 * /A31
IF (VCC) /NIOWRT = /NWR * /A30 * A31 * NCPUSPAC
IF (VCC) /NDISABLE = /FFQAUS * /A15 * /A16 * /A17 * /A18 * /A19
* /A20 * /A21 * NCPUSPAC * /A30 * /A31
+ /NBSACK

IF (/NEPROMCE * TKH) /NDSACK0 = /NEPROMCE * TKH

DESCRIPTION: Erzeugung der Selekt und Disable Signale

OPERATION CODES:
E=ECHO O=PINOUT P=PLOT B=BRIEF
H=HEX L=BHLF N=BNPF Q=QUIT S=SIMULATE
ENTER OPERATION CODE:B

Bild 9. Die PAL-Gleichungen von PA3
    
```

Input	Output
FFQAUS	* 1* P A L *20* VCC
A15	* 2* *19* NDSACK0
A16	* 3* *18* NWR
A17	* 4* *17* A30
A18	* 5* *16* A31
A19	* 6* *15* TKH
A20	* 7* *14* NEPROMCE
A21	* 8* *13* NIOWRT
NBSACK	* 9* *12* NDISABLE
GND	* 10* *11* NCPUSPAC

Bild 10. PAL-Pin-Belegung PA3

mehr möglich ist. Das System arbeitet aber dennoch weiter, nur Floppy-Zugriffe müssen scheitern.

Die Boot-Logik

Bild 8 zeigt die Schaltung. Der 68020 kann mit unterschiedlichen Datenbus-Breiten arbeiten: mit 32-Bit-Datenbus, mit 16-Bit-Datenbus oder mit 8-Bit-Datenbus. Wie breit der aktuelle Datenbus gerade sein soll, das kann man durch die Signale DSACK0 und DSACK1 bestimmen. Dabei kann die Breite von Bus- zu Buszyklus neu festgelegt werden. Die Boot-Logik macht davon Gebrauch. Ein einziges 2764-EPROM genügt nämlich

reichlich für das Boot-Programm, es wäre Verschwendung, vier solcher EPROMs am 32-Bit-Datenbus vorzusehen. Das Boot-EPROM schaltet sich nach dem Reset in den unteren Adressbereich (0 bis 1FFF, bzw. 0 bis 7FFF, um auch 27256-ICs zuzulassen). Ein PAL-Baustein sorgt für die Adressdecodierung. Ein solcher programmierbarer Logik-Baustein kann sehr viele Gatter ersetzen. Er wurde hier zur Schaltungsvereinfachung eingesetzt. Das PAL PA3 erzeugt zum Beispiel das Selekt-Signal für das Boot-EPROM. Ein Flipflop 74LS74 wird dazu nach dem Reset rückgesetzt. Dadurch gelangt ein 0-Signal an Pin1 des PAL-Bausteins. Bild 9 zeigt die PAL-

Gleichungen. In den PAL-Gleichungen sind die Funktionen der Ausgänge in Abhängigkeit von Eingängen definiert. Der Ausgang NEPROMCE führt an das EPROM. Er liegt immer dann auf 0, selektiert also das EPROM, wenn Pin 1 auf 0 liegt, also das Signal FFQAUS, die richtige Adresse anliegt (A16 bis A21 auf 0, sowie A30 und A31 auf 0) und NCPUSPAC auf 0 liegt. NCPUSPAC ist ein Signal, das angibt ob sich die CPU in einem Modus befindet, bei dem die Adressleitungen eine besondere Bedeutung haben. Bild 10 zeigt übrigens die PAL-Belegung, wie sie vom PAL-Assembler erzeugt wird und Bild 11 das Fuse-Diagramm für die, die sich das PAL

```

      11 1111 1111 2222 2222 2233
0123 4567 8901 2345 6789 0123 4567 8901
0 ---- ---- ---- ---- --X- ---X ---- ---- /NEPROMCE*TKH
1 ---- ---- ---- ---- --X- ---X ---- ---- /NEPROMCE*TKH
2 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
3 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
4 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
5 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
6 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
7 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

8 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
9 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
10 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
11 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
12 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
13 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
14 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
15 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

16 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
17 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
18 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
19 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
20 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
21 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
22 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
23 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

24 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
25 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
26 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
27 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
28 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
29 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
30 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
31 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

32 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
33 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
34 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
35 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
36 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
37 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
38 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
39 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

40 ---- ---- ---- ---- ---- ---- ---- ----
41 -X-X -X-- -X-X -X-X -X-- -X-- -X-- -X-- /FFQAUS*/A15*/A16*/A17*/A18*/A1
42 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
43 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
44 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
45 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
46 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
47 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

48 ---- ---- ---- ---- ---- ---- ---- ----
49 ---- ---X ---X ---X ---- ---- ---- ---X- /NWR*/A30*A31*NCPUSPAC
50 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
51 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
52 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
53 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
54 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
55 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

56 ---- ---- ---- ---- ---- ---- ---- ----
57 -X-X -X-- -X-X -X-X -X-- -X-- -X-- -X-- /FFQAUS*/A15*/A16*/A17*/A18*/A1
58 ---- ---- ---- ---- ---- ---- ---- ---X- /NBGACK
59 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
60 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
61 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
62 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX
63 XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

```

LEGEND: X : FUSE NOT BLOWN (L,N,0) OPERATION CODES:
 - : FUSE BLOWN (H,P,1) E=ECHO O=PINOUT P=PLOT B=BRIEF
 H=HEX L=BHLF N=BNPF Q=QUIT
 NUMBER OF FUSES BLOWN = 257 ENTER OPERATION CODE:0 S=SIMULATE

selbst schießen wollen. Das Flipflop kann nun durch einen Speicherzugriff gesetzt werden, womit das EPROM ausgeschaltet wird. Dazu wird der Takteingang des Flipflops mit einem Ausgang des PALs verbunden, der in Bild 9 mit NIOWRT bezeichnet ist. Er wird bei Anlegen einer bestimmten Adresse (A31=1 und A30 = 0, also z. B. auch BFFFFFFC8, siehe Boot-Programm) und einem Schreibsignal (WR) auf 0 gelegt. Damit wird D31 in das Flipflop übernommen. Mit dem Befehl MOVE.B #\$80, \$BFFFFFFC8 kann man also das Flipflop setzen und somit das Boot-EPROM ausblenden.

Das Signal DSACK0 dient zur Busbreitenumschaltung beim 68020. Wenn dieses Signal auf 0 liegt und das Signal DSACK1 auf 1, erwartet die CPU einen 8-Bit-Datenbus. Alle Daten werden dabei auf den Leitungen D31 bis D24 übertragen. Daher sind auch die Q-Ausgänge des EPROMs mit diesen Leitungen verbunden. Das Signal NDSACK0 in der PAL-Gleichung wird noch mit dem Eingang TKH verknüpft. Dieser Eingang erhält sein Signal von einem Schieberegister, das die Aufgabe besitzt, den Zugriff an das EPROM anzupassen, also Wartezyklen einzufügen. Damit können auch sehr langsame EPROMs als Boot-EPROM verwendet werden, ohne daß auf dem System-Bus Wartezyklen eingelegt werden müssen. Der Ausgang DSACK0 ist dabei als offener Kollektor geschaltet. In der PAL-Gleichung steht daher als IF-Bedingung die gleiche Formel wie als Wertzuweisung. Der Ausgang wird in den hochohmigen Zustand geschaltet, wenn diese Gleichung nicht erfüllt ist, und sonst auf 0.

Das Signal NDISABLE in der PAL-Gleichung hat die Aufgabe, bei Zugriff auf das EPROM die internen Treiber und die Zugriffe nach außen auf den Bus zu sperren. Auch bei einem DMA-Zugriff muß das der Fall sein. Daher wird einmal mit der Formel für NEPROMCE, aber auch mit BGACK verknüpft.

Die Steuer-Logik

Bild 12 zeigt die Schaltung. In dieser Schaltung werden alle wichtigen Timing-Signale erzeugt. Dazu werden zwei PAL-Bausteine verwendet. PA1 hat die Aufgabe, die Signale MREQ und IORQ für den Bus zu erzeugen. Jede Bushälfte besitzt ihre eigenen Steuersignale. Me-

Bild 11. Fuse-Diagramm PA3

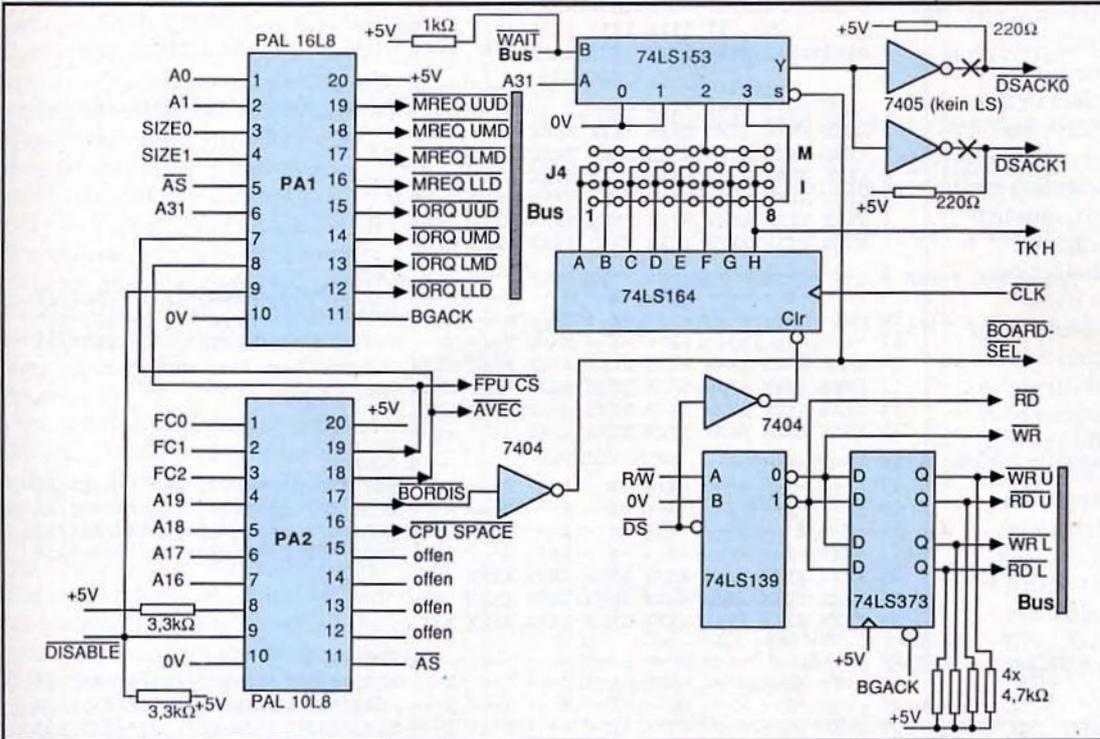


Bild 12. Die Steuer-Logik

PAL16L8

Pa 1 Pal fuer CPU 68020 NDR-Klein-Computer (C) 1986,Rolf-D.Klein

A0 A1 SIZE0 SIZE1 NAS A31 NAVEC NFPUCS NDIS GND
 BGACK IOLLD IOLMD IOUMD IOUUD MRLLD MRLMD MRUMD MRUUD VCC

IF (/BGACK) /MRUUD = NDIS * NFPUCS * NAVEC * /A31 * /NAS * /A0 * /A1
 IF (/BGACK) /MRUMD = NDIS * NFPUCS * NAVEC * /A31 * /NAS * /SIZE0 * /A1
 + NDIS * NFPUCS * NAVEC * /A31 * /NAS * A0 * /A1
 + NDIS * NFPUCS * NAVEC * /A31 * /NAS * /A1 * SIZE1
 IF (/BGACK) /MRLMD = NDIS * NFPUCS * NAVEC * /A31 * /NAS * /A0 * A1
 + NDIS * NFPUCS * NAVEC * /A31 * /NAS
 * /A1 * /SIZE0 * /SIZE1
 + NDIS * NFPUCS * NAVEC * /A31 * /NAS
 * /A1 * SIZE0 * SIZE1
 + NDIS * NFPUCS * NAVEC * /A31 * /NAS
 * A0 * /A1 * /SIZE0
 IF (/BGACK) /MRLLD = NDIS * NFPUCS * NAVEC * /A31 * /NAS
 * A0 * SIZE0 * SIZE1
 + NDIS * NFPUCS * NAVEC * /A31 * /NAS * /SIZE0 * /SIZE1
 + NDIS * NFPUCS * NAVEC * /A31 * /NAS * A0 * A1
 + NDIS * NFPUCS * NAVEC * /A31 * /NAS * A1 * SIZE1
 IF (/BGACK) /IOUUD = NDIS * NFPUCS * NAVEC * A31 * /NAS * /A0 * /A1
 IF (/BGACK) /IOUMD = NDIS * NFPUCS * NAVEC * A31 * /NAS * /SIZE0 * /A1
 + NDIS * NFPUCS * NAVEC * A31 * /NAS * A0 * /A1
 + NDIS * NFPUCS * NAVEC * A31 * /NAS * /A1 * SIZE1
 IF (/BGACK) /IOLMD = NDIS * NFPUCS * NAVEC * A31 * /NAS * /A0 * A1
 + NDIS * NFPUCS * NAVEC * A31 * /NAS
 * /A1 * /SIZE0 * /SIZE1
 + NDIS * NFPUCS * NAVEC * A31 * /NAS
 * /A1 * SIZE0 * SIZE1
 + NDIS * NFPUCS * NAVEC * A31 * /NAS
 * A0 * /A1 * /SIZE0
 IF (/BGACK) /IOLLD = NDIS * NFPUCS * NAVEC * A31 * /NAS
 * A0 * SIZE0 * SIZE1
 + NDIS * NFPUCS * NAVEC * A31 * /NAS * /SIZE0 * /SIZE1
 + NDIS * NFPUCS * NAVEC * A31 * /NAS * A0 * A1
 + NDIS * NFPUCS * NAVEC * A31 * /NAS * A1 * SIZE1

DESCRIPTION: Erzeugt die Signale IORQ und MREQ fuer den BUS, die in UUD UMD LMD und LLD geteilt sind.

OPERATION CODES:

E=ECHO O=PINOUT P=PLOT B=BRIEF
 H=HEX L=BHLF N=BNPF Q=QUIT S=SIMULATE
 ENTER OPERATION CODE:B

Bild 13. PAL-Gleichungen von PA1

A0	* 1*	P A L	*20*	VCC
A1	* 2*	1 6 L 8	*19*	MRUUD
SIZE0	* 3*		*18*	MRUMD
SIZE1	* 4*		*17*	MRLMD
NAS	* 5*		*16*	MRLLD
A31	* 6*		*15*	IOUUD
NAVEC	* 7*		*14*	IOUMD
NFPUCS	* 8*		*13*	IOLMD
NDIS	* 9*		*12*	IOLLD
GND	*10*		*11*	BGACK

OPERATION CODES:

E=ECHO O=PINOUT P=PLOT B=BRIEF
 H=HEX L=BHLF N=BNPF Q=QUIT
 ENTER OPERATION CODE:Q STOP S=SIMULATE

Bild 14. PAL-Pin-Belegung PA1

memory Request dient der Zugriffsteuerung beim Speicher. Das Signal IORQ wird für den Zugriff auf die Peripherie verwendet. Die Signale entsprechen in ihrer Bedeutung den Signalen des Z80, wo ebenfalls Speicher und Peripherie getrennt adressiert werden. Da der Bus an den Z80-Signalen orientiert ist, müssen sie von dieser Schaltung erzeugt werden. Die 68020-CPU unterscheidet verschiedene Zugriffsarten nach Byte, Wort, Langwort und 3/4 Langwort. Dazu liefert sie die Signale SIZE0 und SIZE1. Zusammen mit den Adressen A0 und A1 lassen sich daraus die Steuersignale für die Bushälften errechnen. Bild 13 zeigt die PAL-Gleichungen. Spätestens hier wird klar, daß eine diskrete Logik eine Unmenge von Gattern erfordert hätte. Die Ausgangssignale des PAL-Bausteines können durch das Signal BGACK in den Tri-State-Zustand geschaltet werden, um z. B. bei einem DMA-Zugriff nicht zu stören. Speicher- und Peripheriezugriff werden durch die Adresse A31 unterschieden. Wenn A31 auf 0 liegt, handelt es sich um einen Speicherzugriff, also werden die Signale MREQ aktiv, sonst um einen IO-Zugriff, und die Signale IORQ werden aktiv. Zugegriffen wird aber in jedem Fall nur dann, wenn NDIS auf 1 liegt, also das Disable-Signal 1 ist, NFPUCS auf 1 liegt, also die FPU nicht angesprochen wird, NAVEC auf 1 ist, also kein Interrupt-Zugriff vorliegt und NAS auf 0 liegt, also der Adreß-Strobe aktiv ist, und eine gültige Adresse anliegt.

Die Auswahl der einzelnen Bereiche (UUD..LLD) erfolgt durch zusätzliche Verknüpfung mit den Leitungen A0, A1, SIZE0 und SIZE1. Im Handbuch des 68020 ist die Verknüpfungstabelle anhand einer diskreten Schaltung ausführlich erklärt. Bild 14 zeigt noch die Pinbelegung und Bild 15 die Fuse-Tabelle von PA1.

Eine weitere wichtige Funktion wird durch PA2 erreicht. Es besitzt die Aufgabe, die Dekodierung des Selekt-Signals für die FPU und für den Autovektor vorzunehmen. Ferner werden die Signale für die interne Bussteuerung erzeugt. Bild 16 zeigt die PAL-Gleichungen. Die CPU zeigt Spezial-Zugriffe durch die Signale FC0 bis FC2 an. Wenn alle auf 1 liegen, wird ein sogenannter CPU-SPACE-Zugriff durchgeführt, also ein Zugriff mit einem Adreßraum, der speziell der

	0123	4567	8901	2345	6789	0123	4567	8901	
0	----	----	----	----	----	----	----	----	---X /BGACK
1	-X-X	-----	-----	-X--	-X--	X---	X---	X---	NDIS*NFPUCS*NAVEC*/A31*/NAS*/A0
2	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
3	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
4	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
5	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
6	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
7	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
8	----	----	----	----	----	----	----	----	---X /BGACK
9	-X--	-X--	-----	-X--	-X--	X---	X---	X---	NDIS*NFPUCS*NAVEC*/A31*/NAS*/SI
10	-XX-	-----	-----	-X--	-X--	X---	X---	X---	NDIS*NFPUCS*NAVEC*/A31*/NAS*A0*
11	-X--	-----	X---	-X--	-X--	X---	X---	X---	NDIS*NFPUCS*NAVEC*/A31*/NAS*/A1
12	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
13	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
14	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
15	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
16	----	----	----	----	----	----	----	----	---X /BGACK
17	X--X	-----	-----	-X--	-X--	X---	X---	X---	NDIS*NFPUCS*NAVEC*/A31*/NAS*/A0
18	-X--	-X--	-X--	-X--	-X--	X---	X---	X---	NDIS*NFPUCS*NAVEC*/A31*/NAS*/A1
19	-X--	X---	X---	-X--	-X--	X---	X---	X---	NDIS*NFPUCS*NAVEC*/A31*/NAS*/A1
20	-XX-	-X--	-----	-X--	-X--	X---	X---	X---	NDIS*NFPUCS*NAVEC*/A31*/NAS*A0*
21	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
22	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
23	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
24	----	----	----	----	----	----	----	----	---X /BGACK
25	--X-	X---	X---	-X--	-X--	X---	X---	X---	NDIS*NFPUCS*NAVEC*/A31*/NAS*A0*
26	----	-X--	-X--	-X--	-X--	X---	X---	X---	NDIS*NFPUCS*NAVEC*/A31*/NAS*/SI
27	X-X-	-----	-----	-X--	-X--	X---	X---	X---	NDIS*NFPUCS*NAVEC*/A31*/NAS*A0*
28	X---	-----	X---	-X--	-X--	X---	X---	X---	NDIS*NFPUCS*NAVEC*/A31*/NAS*A1*
29	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
30	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
31	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
32	----	----	----	----	----	----	----	----	---X /BGACK
33	-X-X	-----	-----	-X--	X---	X---	X---	X---	NDIS*NFPUCS*NAVEC*A31*/NAS*/A0*
34	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
35	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
36	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
37	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
38	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
39	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
40	----	----	----	----	----	----	----	----	---X /BGACK
41	-X--	-X--	-----	-X--	X---	X---	X---	X---	NDIS*NFPUCS*NAVEC*A31*/NAS*/SIZ
42	-XX-	-----	-----	-X--	X---	X---	X---	X---	NDIS*NFPUCS*NAVEC*A31*/NAS*A0*/
43	-X--	-----	X---	-X--	X---	X---	X---	X---	NDIS*NFPUCS*NAVEC*A31*/NAS*/A1*
44	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
45	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
46	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
47	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
48	----	----	----	----	----	----	----	----	---X /BGACK
49	X--X	-----	-----	-X--	X---	X---	X---	X---	NDIS*NFPUCS*NAVEC*A31*/NAS*/A0*
50	-X--	-X--	-X--	-X--	X---	X---	X---	X---	NDIS*NFPUCS*NAVEC*A31*/NAS*/A1*
51	-X--	X---	X---	-X--	X---	X---	X---	X---	NDIS*NFPUCS*NAVEC*A31*/NAS*/A1*
52	-XX-	-X--	-----	-X--	X---	X---	X---	X---	NDIS*NFPUCS*NAVEC*A31*/NAS*A0*/
53	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
54	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
55	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
56	----	----	----	----	----	----	----	----	---X /BGACK
57	--X-	X---	X---	-X--	X---	X---	X---	X---	NDIS*NFPUCS*NAVEC*A31*/NAS*A0*S
58	----	-X--	-X--	-X--	X---	X---	X---	X---	NDIS*NFPUCS*NAVEC*A31*/NAS*/SIZ
59	X-X-	-----	-----	-X--	X---	X---	X---	X---	NDIS*NFPUCS*NAVEC*A31*/NAS*A0*A
60	X---	-----	X---	-X--	X---	X---	X---	X---	NDIS*NFPUCS*NAVEC*A31*/NAS*A1*S
61	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
62	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	
63	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	XXXX	

LEGEND: X : FUSE NOT BLOWN (L,N,0) - : FUSE BLOWN (H,P,1)

NUMBER OF FUSES BLOWN = 840

OPERATION CODES:

E=ECHO O=PINOUT P=PLOT B=BRIEF
H=HEX L=BHLF N=BNPF Q=QUIT S=SIMULATE
ENTER OPERATION CODE:0

Bild 15. Fuse-Diagramm PA1

CPU gehört. Der PAL-Baustein erzeugt das Signal NCPUSPAC, das diesen Zustand auch nach außen an andere Schaltungsteile weitergibt, denn für diesen Zugriff besitzt der Adreßbus eine besondere Bedeutung und darf nicht wie gewohnt dekodiert werden. Die CPU kann mit dieser Zugriffsart zum Beispiel Co-Prozessoren adressieren. Welcher Co-Prozessor dabei gemeint ist, wird durch die Adreßleitungen A16 bis A19 mitgeteilt. Daher wird im PAL eine weitere Verknüpfung durchgeführt: Wenn A19=0, A18=0, A17=1 und A16=0 sind, so erfolgt ein Zugriff auf die FPU (Floating Point Unit). Dieser Zugriff wird von der CPU automatisch gesteuert, so daß die FPU wie eine Befehlsweiterung im Instruktionssatz aussieht. Die CPU versteht also mehr Befehle. Auf unserer Karte wird nur der Autovektor-Mechanismus zur Interrupt-Bearbeitung eingesetzt, der die Erzeugung spezieller Interrupt-Vektoren auf dem Bus überflüssig macht. Dadurch ist es möglich, den Bus für unterschiedliche Prozessoren kompatibel zu halten. Eine Interrupt-Bestätigung erhält man, wenn die Adreßleitungen A19 bis A16 auf 1 liegen und FC0 bis FC2 ebenfalls. Dann wird hier das Signal NAVEC erzeugt, das direkt an die CPU geleitet wird. Dadurch wird eine besondere Bedeutung für die Leitungen IPL0 bis IPL1 der CPU programmiert. Die CPU führt dann die Interrupts auf feste Adressen. Ein weiteres Signal, NBOARDIS, wird vom PAL-Baustein erzeugt. Es hat die Aufgabe, die Datenbustreiber und einige andere Schaltungsteile zu sperren, wenn entweder ein FPU-Zugriff vorliegt oder das Signal NDISABLE (Disable) anliegt. DISABLE kommt von PA3 und wird erzeugt, wenn ein DMA-Zugriff oder ein Zugriff auf das Boot-EPROM vorliegt. Bild 17 zeigt die Pin-Belegung, Bild 18 die Fuse-Tabelle von PA3. Nun zur restlichen Steuer-Logik. Ein Dekoder 74LS139 übernimmt die Erzeugung der Signale RD und WR. RD tritt bei einem Lese-Vorgang auf, und WR bei einem Schreib-Vorgang. Der Baustein 74LS373 hat die Aufgabe diese Signale zu puffern. Dabei bekommt jede Bushälfte ein eigenes Signal, um das Fan Out zu erhöhen. Diese Signale WR U, WR L, RD U und RD L, deren U- und L-Hälfte jeweils zur gleichen Zeit aktiv werden, sind von den internen Signalen RD und WR abgeleitet. Der Baustein 74LS373 wird von BGACK gesteuert, wie die anderen 74LS373-Bausteine, damit bei einem DMA-Zugriff die Signale RD und WR auf dem Bus frei zur Verfügung stehen. Pull-Up-Widerstände sorgen dafür,

```

ENTER FILENAME (WITH EXTENSION) ---> PA2.PAL
PAL10L8

Pa 2 Pal fuer CPU 68020 NDR-Klein-Computer (C) 1986,Rolf-D.Klein

FC0 FC1 FC2 A19 A18 A17 A16 NC NDISABLE GND
NAS NC NC NC NC NCPUSPAC NBOARDIS NAVEC NFPUCS VCC

/NFPUCS = FC0 * FC1 * FC2 * /A19 * /A18 * A17 * /A16
/NAVEC = FC0 * FC1 * FC2 * A19 * A18 * A17 * A16 * /NAS
/NBOARDIS = FC0 * FC1 * FC2 * /A19 * /A18 * A17 * /A16
+ /NDISABLE
/NCPUSPAC = FC0 * FC1 * FC2

DESCRIPTION: Erzeugung diverser Select-Signale
    
```

OPERATION CODES:

E=ECHO O=PINOUT P=PLOT B=BRIEF
H=HEX L=BHLF N=BNPF Q=QUIT S=SIMULATE
ENTER OPERATION CODE:P

Bild 16.
PAL-Gleichungen
von PA2

	*****	*****	
	*	* *	*
FC0	****	P A L	****
	* 1*		*20* VCC
	****		****
	*	1 0 L 8	*
FC1	****		****
	* 2*		*19* NFPUCS
	****		****
	*		*
FC2	****		****
	* 3*		*18* NAVEC
	****		****
	*		*
A19	****		****
	* 4*		*17* NBOARDIS
	****		****
	*		*
A18	****		****
	* 5*		*16* NCPUSPAC
	****		****
	*		*
A17	****		****
	* 6*		*15* NC
	****		****
	*		*
A16	****		****
	* 7*		*14* NC
	****		****
	*		*
NC	****		****
	* 8*		*13* NC
	****		****
	*		*
NDISABLE	****		****
	* 9*		*12* NC
	****		****
	*		*
GND	****		****
	10		*11* NAS
	****		****
	*		*

OPERATION CODES:

E=ECHO O=PINOUT P=PLOT B=BRIEF
H=HEX L=BHLF N=BNPF Q=QUIT S=SIMULATE
ENTER OPERATION CODE:Q STOP

Bild 17.
PAL-PIN-
Belegung PA2

daß nach dem Umschalten kein undefinierter Zustand auf dem Bus auftritt, falls die DMA-Schaltung die Signale erst nach einiger Verzögerungszeit belegt.

Das Tempo muß passen

Mit einem Schieberegister 74LS164 und einem Multiplexer 74LS153 werden Wartezyklen paßgerecht erzeugt. In unserer Schaltung kann man Speicher- und IO-Zyklen getrennt einstellen. Von Position 1 bis 8 wächst die Zahl der eingestellten Wartezyklen. Die FLO2-Baugruppe benötigt ca. 3 bis 4 Wartezyklen bei 16-MHz-Betrieb. Daher sollte man die IO-Zyklen auf diesen Wert stellen (Brücke I auf 4). Beim Speicher genügt ein Wartezyklus (Brücke M auf 1). Der Multiplexer hat die Aufgabe die jeweiligen Wartezyklen aufzuschalten. Dazu wird an den A-Eingang das Signal A31 geführt. Es unterscheidet Speicher von Peripherie. Auf den Eingang B ist zusätzlich die Wait-Leitung des Bus geführt. Damit kann auch eine externe Schaltung ein Wartesignal erzeugen. Die Warte-Steuerung erfolgt über DSACK0 und DSACK1. Da außen immer ein 32-Bit-Bus vorhanden ist, werden beide Signale über die Gatter 7405 gleichzeitig auf 0 gelegt. Nach Beginn eines Bus-Zyklus wartet die CPU, bis DSACK0 oder DSACK1 oder beide auf 0 gehen. Durch das Schieberegister wird ein durchlaufendes 1-Signal an seinen Ausgängen erzeugt, sobald das Signal DS (Datastrobe) den Zugriffbeginn mitteilt. Über den Multiplexer gelangt dann das Speicher- oder Peripherie-Signal an die Gatter im 7405. Wenn von außen WAIT angelegt wird, also die Leitung WAIT am Bus auf 0 geht, so schaltet der Multiplexer auf die Eingänge 0 oder 1 um und erhält von dort ein 0-Signal. DSACK0 und DSACK1 bleiben damit auf 1 und die CPU wartet. Das Schieberegister arbeitet normal weiter, und DSACK0 und DSACK1 werden nach Ende des Wait-Signales sofort auf 0 gelegt, falls das Schieberegister ebenfalls schon mit seinen Wartezyklen durch ist. Der Multiplexer wird zusätzlich durch das Signal BOARDSEL gesteuert, denn wenn die CPU auf die FPU zugreift, so erzeugt diese ihre eigenen DSACK-Signale. Bei einem Zugriff auf das Boot-EPROM wird der Multiplexer über das BOARDSEL-Signal ebenfalls gesperrt, denn das EPROM erzeugt ja ein eigenes

	11	1111	1111	2222	2222	2233	
0123	4567	8901	2345	6789	0123	4567	8901
0	X-X-	X-00	-X00	-X00	X-00	-X00	--00 ---- FC0*FC1*FC2*/A19*/A18*A17*/A16
1	XXXX	XX00	XX00	XX00	XX00	XX00	XXXX
2	0000	0000	0000	0000	0000	0000	0000
3	0000	0000	0000	0000	0000	0000	0000
4	0000	0000	0000	0000	0000	0000	0000
5	0000	0000	0000	0000	0000	0000	0000
6	0000	0000	0000	0000	0000	0000	0000
7	0000	0000	0000	0000	0000	0000	0000
8	X-X-	X-00	X-00	X-00	X-00	-X00	--00 ---X FC0*FC1*FC2*A19*A18*A17*A16*/NA
9	XXXX	XX00	XX00	XX00	XX00	XX00	XXXX
10	0000	0000	0000	0000	0000	0000	0000
11	0000	0000	0000	0000	0000	0000	0000
12	0000	0000	0000	0000	0000	0000	0000
13	0000	0000	0000	0000	0000	0000	0000
14	0000	0000	0000	0000	0000	0000	0000
15	0000	0000	0000	0000	0000	0000	0000
16	X-X-	X-00	-X00	-X00	X-00	-X00	--00 ---- FC0*FC1*FC2*/A19*/A18*A17*/A16
17	----	--00	--00	--00	--00	--00	-X-- /NDISABLE
18	0000	0000	0000	0000	0000	0000	0000
19	0000	0000	0000	0000	0000	0000	0000
20	0000	0000	0000	0000	0000	0000	0000
21	0000	0000	0000	0000	0000	0000	0000
22	0000	0000	0000	0000	0000	0000	0000
23	0000	0000	0000	0000	0000	0000	0000
24	X-X-	X-00	--00	--00	--00	--00	---- FC0*FC1*FC2
25	XXXX	XX00	XX00	XX00	XX00	XX00	XXXX
26	0000	0000	0000	0000	0000	0000	0000
27	0000	0000	0000	0000	0000	0000	0000
28	0000	0000	0000	0000	0000	0000	0000
29	0000	0000	0000	0000	0000	0000	0000
30	0000	0000	0000	0000	0000	0000	0000
31	0000	0000	0000	0000	0000	0000	0000
32	XXXX	XX00	XX00	XX00	XX00	XX00	XXXX
33	XXXX	XX00	XX00	XX00	XX00	XX00	XXXX
34	0000	0000	0000	0000	0000	0000	0000
35	0000	0000	0000	0000	0000	0000	0000
36	0000	0000	0000	0000	0000	0000	0000
37	0000	0000	0000	0000	0000	0000	0000
38	0000	0000	0000	0000	0000	0000	0000
39	0000	0000	0000	0000	0000	0000	0000
40	XXXX	XX00	XX00	XX00	XX00	XX00	XXXX
41	XXXX	XX00	XX00	XX00	XX00	XX00	XXXX
42	0000	0000	0000	0000	0000	0000	0000
43	0000	0000	0000	0000	0000	0000	0000
44	0000	0000	0000	0000	0000	0000	0000
45	0000	0000	0000	0000	0000	0000	0000
46	0000	0000	0000	0000	0000	0000	0000
47	0000	0000	0000	0000	0000	0000	0000
48	XXXX	XX00	XX00	XX00	XX00	XX00	XXXX
49	XXXX	XX00	XX00	XX00	XX00	XX00	XXXX
50	0000	0000	0000	0000	0000	0000	0000
51	0000	0000	0000	0000	0000	0000	0000
52	0000	0000	0000	0000	0000	0000	0000
53	0000	0000	0000	0000	0000	0000	0000
54	0000	0000	0000	0000	0000	0000	0000
55	0000	0000	0000	0000	0000	0000	0000
56	XXXX	XX00	XX00	XX00	XX00	XX00	XXXX
57	XXXX	XX00	XX00	XX00	XX00	XX00	XXXX
58	0000	0000	0000	0000	0000	0000	0000
59	0000	0000	0000	0000	0000	0000	0000
60	0000	0000	0000	0000	0000	0000	0000
61	0000	0000	0000	0000	0000	0000	0000
62	0000	0000	0000	0000	0000	0000	0000
63	0000	0000	0000	0000	0000	0000	0000

LEGEND: X : FUSE NOT BLOWN (L,N,0) - : FUSE BLOWN (H,P,1)
 0 : PHANTOM FUSE (L,N,0) O : PHANTOM FUSE (H,P,1)

NUMBER OF FUSES BLOWN = 134

OPERATION CODES:

E=ECHO O=PINOUT P=PLOT B=BRIEF
 H=HEX L=BHLF N=BNPF Q=QUIT S=SIMULATE
 ENTER OPERATION CODE:0

Bild 18. Fuse-Diagramm PA2

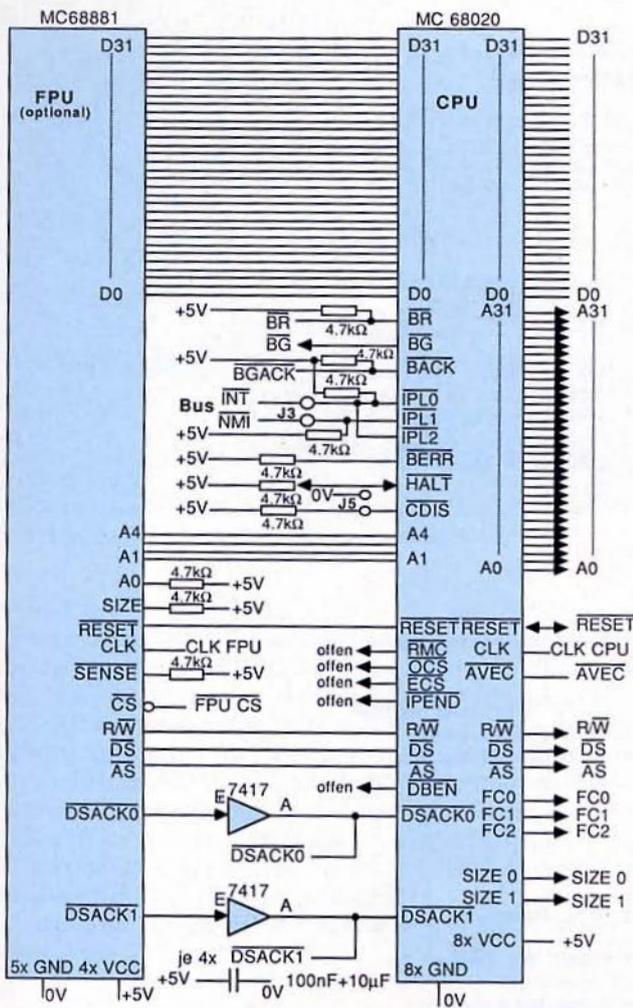


Bild 19. CPU und FPU im Verbund



Bild 20. Die CPU-Pin-Belegung

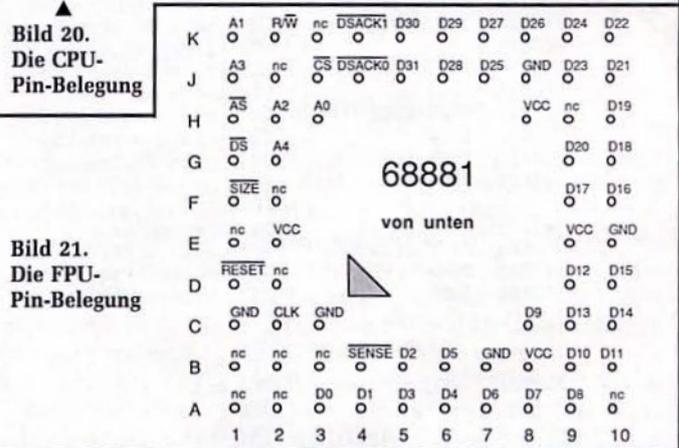


Bild 21. Die FPU-Pin-Belegung

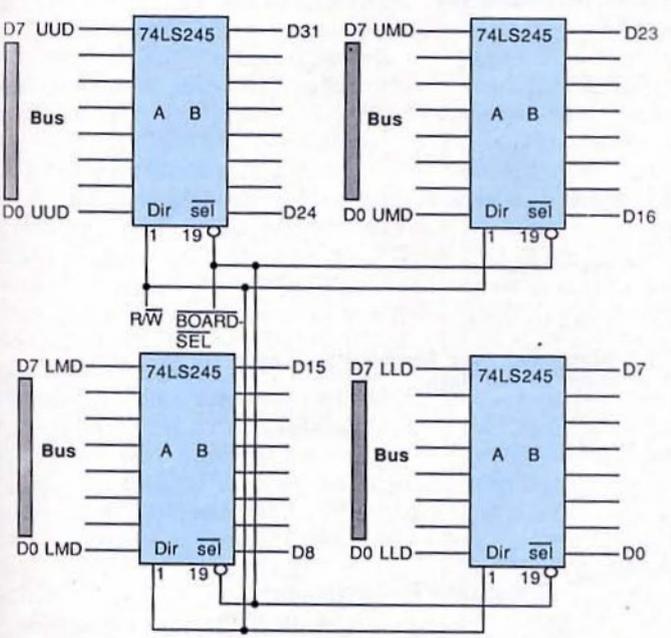


Bild 22. Die Datenbus-Treiber

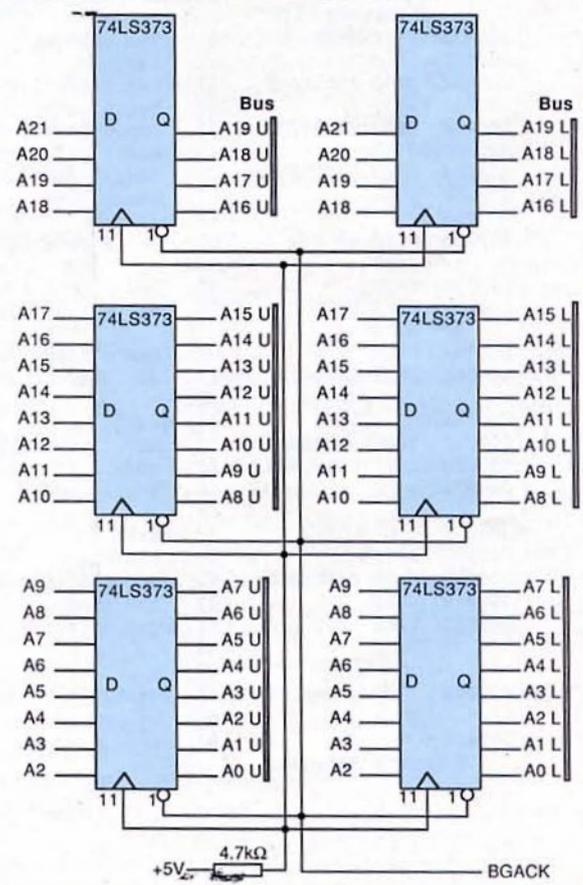


Bild 23. Adreß-Bus-Treiber

```

...A68K 4.0a 06/13/83 ...Run on 09/05/85
1. *****
2. * Boot fuer 68020 auf der Baugruppe *
3. *
4. * CPU 68020/32 V1.0 850710 RDK *
5. * V 1.1, 860315 mit RAM Pruefung *
6. * und Erkennung *
7. *****
8.
000000 9. section 0
10.
000000 00008FFE 11. dc.l $8ffe * Vorlaeufiger Stack.
000004 '00000008 12. dc.l start
13.
000008 14. start:
000008 41F9 00008070 15. lea $8070,a0 * Ziel, (liegt ggf in ausbuf,einbuf GRUPROG).
16. * dort 264 Bytes Platz. (rev 1.1).
00000E 2448 17. movea.l a0,a2 * Dort Start merken
000010 18. fehlerwdh: * Einsprung im Fehlerfall
000010 43FA 0020 19. lea anf(pc),a1 * Quelle im Boot-EPROM.
000014 363C 0053 20. move #ende-anf-1,d3
000018 21. transport:
000018 1019 22. move.b (a1)+,d0
00001A 10C0 23. move.b d0,(a0)+
00001C B028 FFFF 24. cmp.b -(a0),d0
000020 6606 25. bne.s errorskip
000022 51CB FFF4 26. dbra d3,transport
000026 4ED2 27. jmp (a2) * Gueltige Startadresse
28.
000028 29. errorskip:
000028 D5FC 00002000 30. adda.l #$2000,a2 * in 8K Schritten suchen.
00002E 204A 31. movea.l a2,a0 * neue Zieladresse
000030 60DE 32. bra.s fehlerwdh * dort neu versuchen.
33.
000032 34.
35.
000032 36. anf:
37.
000032 38. boot:
000032 13FC 0080 39. move.b #$80,$BFFFFFFC8 * Boot loeschen
BFFFFFFC8
00003A 41F8 0400 40. lea $400,a0 * Start der Suche
00003E 41. loop:
00003E 0C90 A55A8002 42. cmp.l #$a55a8002,(a0) * Anwender-Kennung
000044 6732 43. beq.s chess
000046 0C90 5AA58001 44. cmp.l #$5aa58001,(a0) * Grundprog. Kennung
00004C 6708 45. beq.s gefunden
00004E D1FC 00000400 46. adda.l #$400,a0 * 1K Seitengrenzen
000054 60E8 47. bra.s loop
000056 48. gefunden:
000056 0C68 6000 49. cmp #$6000,$20(a0)
0020
00005C 66E0 50. bne.s loop
00005E 0C68 6000 51. cmp #$6000,$24(a0)
0024
000064 66D8 52. bne.s loop
000066 4EE8 0024 53. jmp $24(a0) * Start auch wenn in RAM
54.
00006A 55. goon:
00006A 20BC A55A8002 56. move.l #$a55a8002,(a0) * restaurieren
000070 D1FC 00001000 57. adda.l #$1000,a0
000076 60C6 58. bra.s loop
59.
000078 60. chess:
000078 4290 61. clr.l (a0) * Einsprung fuer Anwender Software.
00007A 0C90 A55A8002 62. cmp.l #$a55a8002,(a0) * muss in ROM sein.
000080 66E8 63. bne goon
000082 4EE8 0004 64. jmp $4(a0) * und starten
65.
000086 66.
67. ende:
000086 68.
69. end
0 Errors detected

```

Bild 24. Das Boot-Programm

Minimalsystem:	1x Baugruppe CPU 68020 + Boot-EPROM 2x großer Bus 4x ROA64 + 4x RAM8K + 8x EPROM64K (Gru) 1x KEY + ASCII-Tastatur 1x GDP64 + Bildschirm
CP/M-Ausbau:	dazu: 4x DYN 256K 1x FLO2 1x Floppy-Laufwerk (besser 2) 1x Betriebssystem CP/M
Erweiterungen:	CENT+ IOE für Parallel-Drucker SER für serielle Schnittstelle HCOPY für Maus und/oder Graphikausgabe FESTCON für Winchester-Anschluß usw.

Bild 25. Das sind die verschiedenen Ausbaustufen

DSACK-Signal (wobei nur DSACK0 erzeugt wird, da das EPROM mit einem 8-Bit-Datenbus arbeitet). Die beiden Signale DSACK0 und DSACK1 sind übrigens über einen sehr niederohmigen Widerstand (220 Ohm) nach +5 V verbunden um die Schaltung für 16 MHz tauglich zu machen. Sind die Widerstände nämlich zu groß, so steigen die Signale DSACK0 und DSACK1 zu langsam an, und der nächste Bus-Zyklus erhält noch das 0-Signal als Eingangswert und arbeitet dann ohne Wartezyklen.

Die CPU und FPU

Bild 19 zeigt die Beschaltung der CPU mit FPU. Die FPU ist fast völlig parallel zur CPU geschaltet. Nur die Signale A0, SIZE, sowie CLK, SENSE und CS liegen nicht parallel. A0 und SIZE bestimmen die Datenbusbreite, die hier auf 32 Bit eingestellt ist. SENSE ist von der FPU aus ständig auf Masse gelegt. Damit könnte man über einen Port abfragen ob die FPU im Sockel steckt, was hier jedoch nicht getan wird. Die Signale CLK, also der Takt, kommen von der Taktlogik und sind dort über einen 47-Ohm-Widerstand mit dem gemeinsamen Takt verbunden. Übrigends ist es möglich, FPU und CPU unterschiedlich zu takten. Das Signal CS ist das Auswahl-Signal an die FPU und wird durch PA1 erzeugt. Die FPU erzeugt noch zwei Rückmeldesignale, DSACK0 und DSACK1, die über nicht-invertierende Treiber mit den entsprechenden Eingängen der CPU verbunden sind. Darüber werden Datentransfer-Ende und die Datenbusbreite signalisiert. Bei der CPU sind die Signale IPL0 bis IPL2 bedeutsam. Sie dienen der Inter-

rupt-Anforderung. IPL0 und IPL2 sind zusammenschaltet, damit man vom Bus her das gleiche Verhalten wie beim 68008 bekommt, zumal nur zwei Interruptleitungen auf dem Bus zur Verfügung stehen. Will man einen nicht-maskierbaren Interrupt benutzen, so muß man die Brücke J3 setzen. Dann hat man allerdings nur einen Interrupt-Eingang. Die Leitung CDIS kann über die Brücke J5 auf 0 gelegt werden. Damit ist der interne Cache-Speicher gesperrt. Wenn man den Cache-Speicher verwenden will, muß man ihn aber erst per Befehl einschalten (MOVE.L #1,D0 ; MOVEC D0,CACR). Zusätzlich muß die Brücke offen sein.

Die CPU besitzt acht Masse-Anschlüsse und acht Versorgungs-Spannungseingänge, die alle angeschlossen sein müssen. Die FPU besitzt fünf Masse- und vier +5-V-Eingänge. Einige Entstörkondensatoren sollte man zusätzlich in der Nähe der CPU und FPU anbringen (wie bereits im Layout berücksichtigt). Bild 20 zeigt die Pin-Belegung der CPU und Bild 21 die Pin-Belegung der FPU (jeweils von unten gezeichnet). Bild 22 zeigt die Schaltung der Datenbus-Treiber. Durch das Signal R/W wird deren Richtung gesteuert und die Freigabe erfolgt mit dem Signal BOARDSEL. Bild 23 zeigt die Schaltung der Adressbus-Treiber. Für jede Bushälfte sind eigene Treiber vorgesehen, um das Fan Out zu erhöhen und lange Leitungen auf der Leiterplatte zu puffern. Die Bausteine 74LS373 werden durch das Signal BGACK freigegeben, so daß bei einem DMA-Zugriff der Bus zur Verfügung steht. Der Adreßbus ist auf besondere Weise verdrahtet. Die Leitung A2 der CPU

führt auf die Busleitung Leitung A0. Alles ist also um zwei Adressen verschoben. Da mit einem 32-Bit-Datenbus gearbeitet wird, aber beim 68020 auch Byte-Adressierung möglich ist, werden die CPU-internen Leitungen A0 und A1 in entsprechende Select-Signale (PA1) umgewandelt. Beim NDR-Klein-Computer kann man mit dem Original-Bus so 4 MByte mit dem 68020 adressieren und im Gegensatz zum 68008 (1 MByte) ist der IO-Bereich eigentlich auch genauso groß (4 MByte). Jedoch decodieren die normalen IO-Baugruppen des NDR-Klein-Systems nur die unteren 8 Bit des Adressbus, so daß man „nur“ 1024 IO-Adressen bekommt (auf jedem Busviertel 256 IO-Adressen).

Das Boot-Programm

Bild 24 zeigt das Programm. Zum Betrieb der Baugruppe benötigt man auch das Boot-EPROM. Es hat die Aufgabe, nach dem Grundprogramm zu suchen und dieses zu starten. Durch das Boot-Prinzip ist es möglich, auch auf Adresse 0 RAM zu betreiben, wie man es für den Betrieb von CP/M-68k benötigt. Das Programm kopiert zunächst das eigentliche Suchprogramm in einen RAM-Bereich. Dabei wird bei Adresse \$8070 begonnen nach RAM zu suchen. Die Adresse wurde so gewählt, weil sie in das lokale RAM des Grundprogramms fallen kann (Version 4.3 auf \$8000 Version 5.x auf \$10000). Dann wird da nur ein unbedeutender Puffer überschrieben. Damit ist ein zerstörungsfreier Warmstart nach einem erneuten Reset möglich. Das Bootprogramm schaltet, sobald es sich im RAM befindet und dort aufgerufen wurde, das Boot-EPROM ab und beginnt die Suche bei Adresse \$400 in 1-KWort-Schritten. Das Grundprogramm besitzt ein Erkennungsmuster (\$55AA8001) mit dem das Suchprogramm den Anfang finden kann. Falls das Muster gefunden ist, wird eine Plausibilitätsprüfung durchgeführt und dann wird das Grundprogramm gestartet. Ein weiterer String, der \$a55a8002 lautet, wird ebenfalls gesucht. Mit ihm kann man eigene Anwendersoftware kennzeichnen, die dann auch automatisch gestartet werden kann. Fortsetzung folgt: FPU, CPU und Software

Literatur:

- [1] Mikrocomputer Schritt für Schritt 2. Sonderheft von mc, Franzis-Verlag.
- [2] Motorola: MC68020 32-Bit Microprocessor User's Manual.