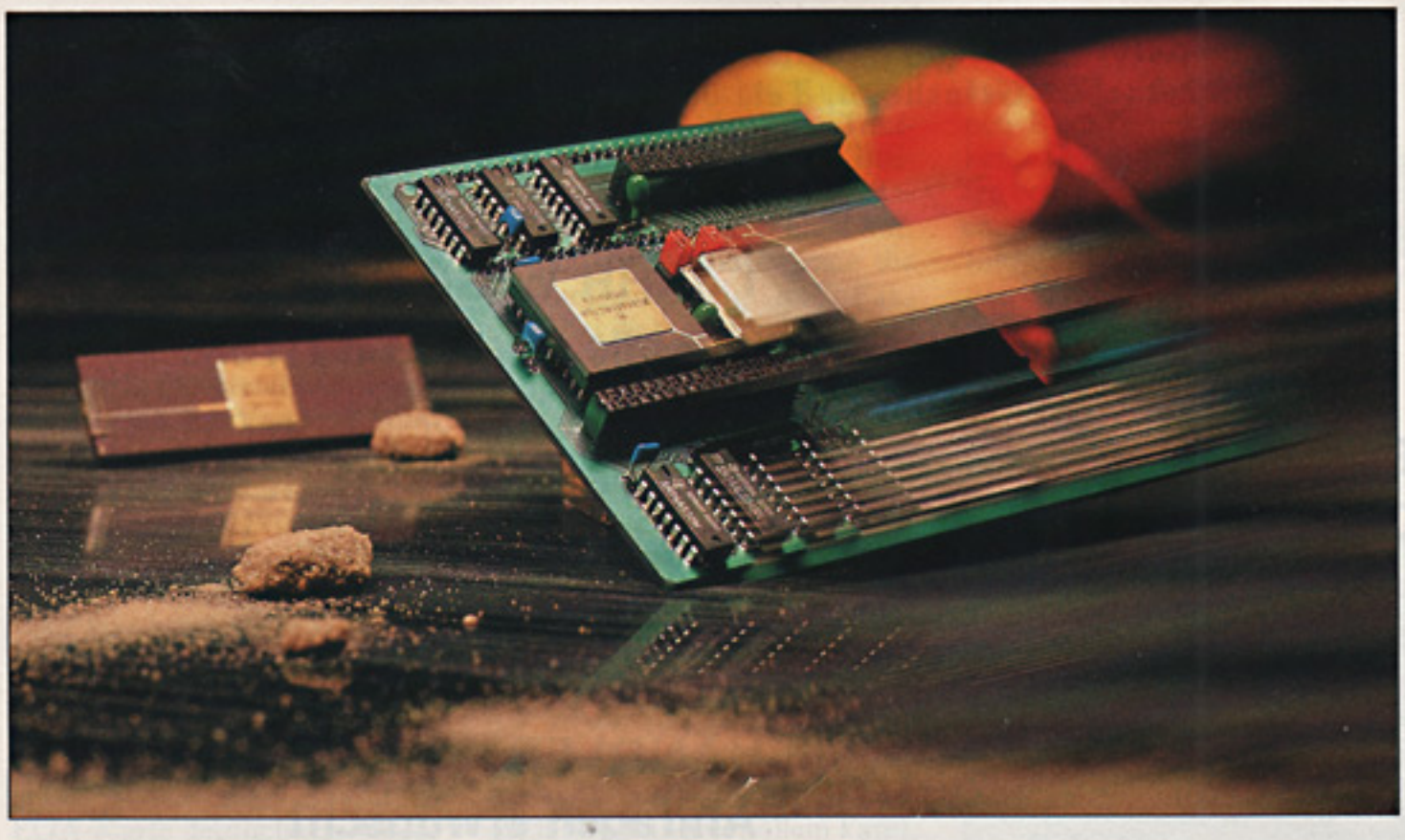


Mehr CPU

PAK-68 – die Prozessor-Austausch-Karte für 68000-Rechner

Johannes Assenbaum

Seit es den 68020, die 32-Bit-CPU aus der 68000-Familie, und den Floating-Point-Coprozessor 68881 gibt, träumt so mancher Computerist von diesen Prozessoren in seinem Rechner. Viele haben es inzwischen zu einem 68000-System gebracht – doch wie den Sprung von 16 auf 32 Bit schaffen? Man will ja nicht (schon wieder) ein teures neues Gerät anschaffen.

Einen 68000-Computer mit dem 'Superchip' 68020 zu tunen, ist durchaus möglich, wenn auch nicht ganz billig, insbesondere dann nicht, wenn man den Arithmetik-Coprozessor 68881 gleich dabeihaben möchte. Aber stellen Sie sich einmal vor, Ihr Rechner wäre 30 bis 40 Prozent oder, bei rechenintensiven Programmen, noch schneller – könnte Sie das nicht reizen?

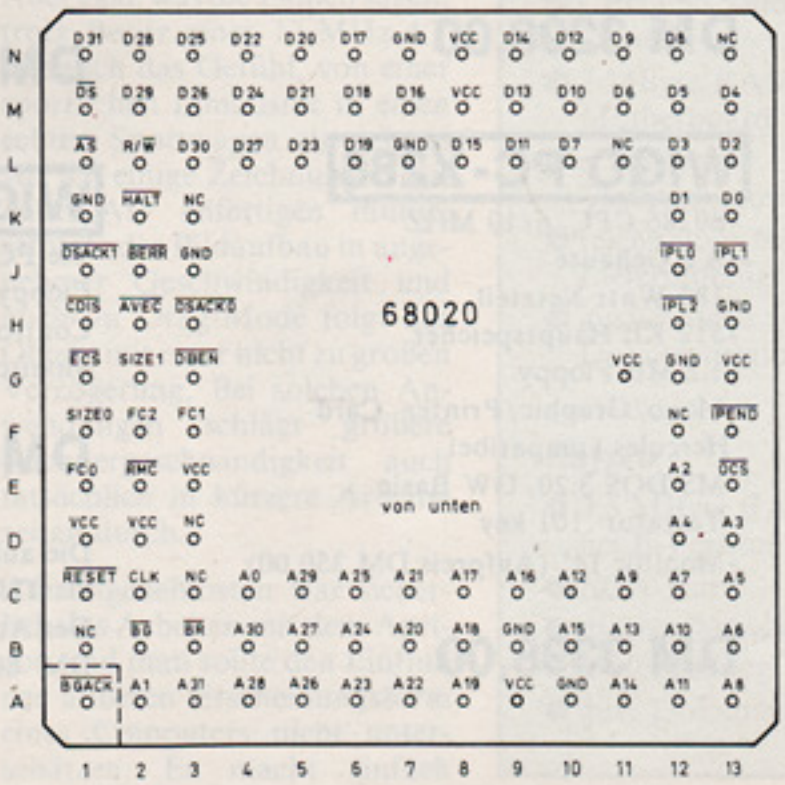
Was man dazu braucht, ist ein Adapter, der in erster Linie die 32-Bit-Busse der Prozessorchips 68020/881 auf den 16-Bit-Daten- und den 24-Bit-Adreßbus des 68000 umsetzt. Daneben gibt es auch bei den Steuersignalen Unterschiede, die ein solcher Adapter auszugleichen hat.

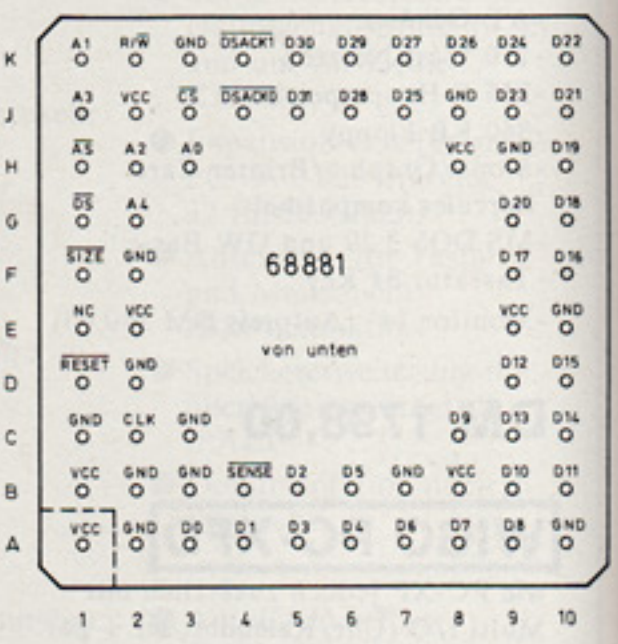
13 Chips

Die c't-Prozessor-Austausch-Karte für 68000-Systeme PAK-68 erledigt dies mit 13 ICs; zwei davon sind die Prozessoren, der Rest LS-, ALS- und AS-TTL-Typen. Eine vierzehnte Fassung nimmt den Quarzoszillator auf, der das Ganze noch etwas auf Trapp bringen kann. Als besonderes Bonbon ist es möglich, auf einer extra Karte ein 32 Bit breites RAM hinzuzufügen, das dann erst recht die Post abgehen läßt.

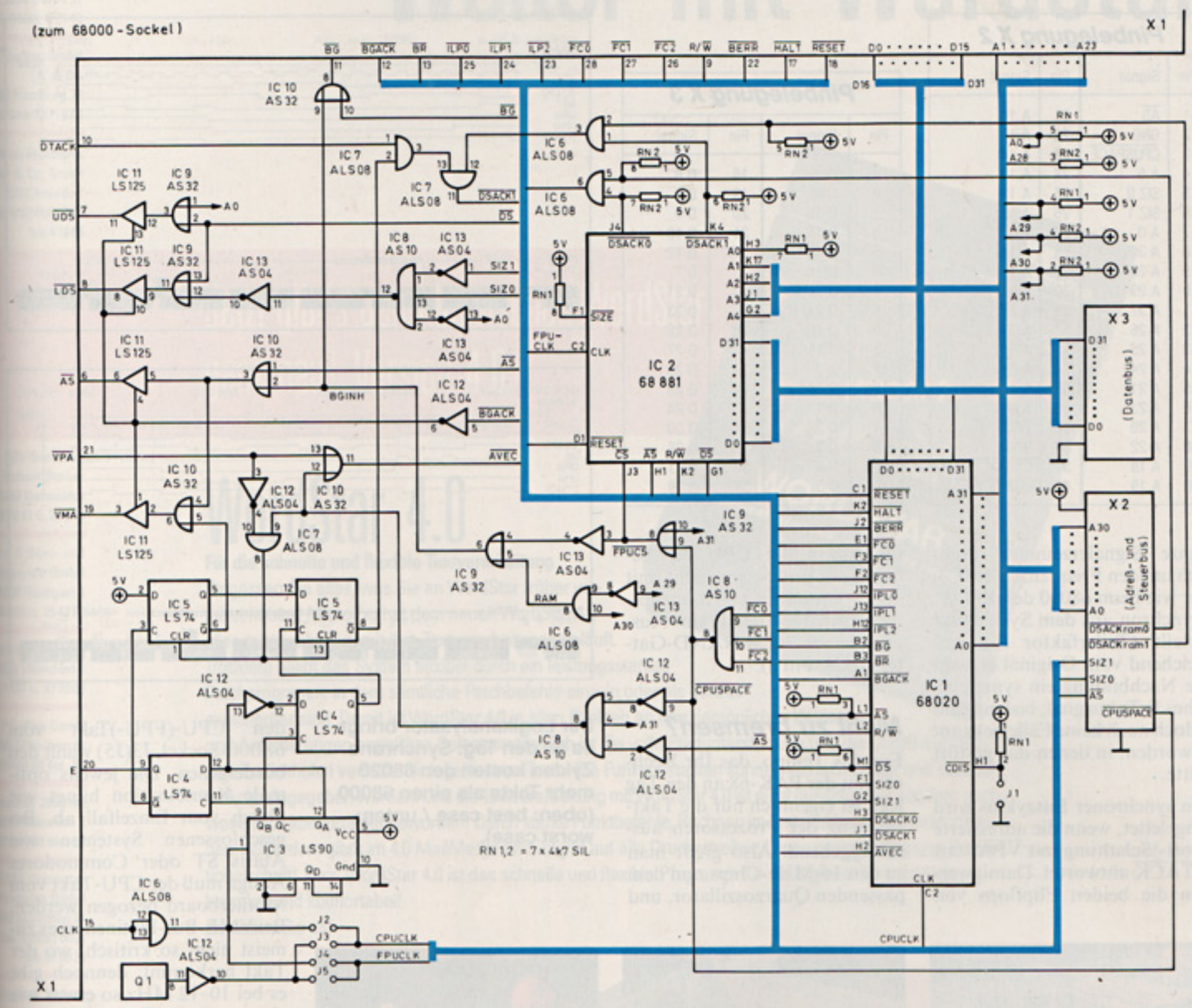
Um CPU und FPU zusammenzubringen, bedarf es im Prinzip nur einiger weniger Gatterfunktionen für die Chipselektion der FPU. Bei der PAK-68 wird dazu das aus den Function-Codes der CPU ($\overline{FC0...2}$) gewonnene $\overline{CPUSPACE}$-Signal mit dem höchstwertigen Adreßbit invertiert-AND-verknüpft (entspricht OR). Im gegebenen Kontext reicht dies für eine eindeutige Unterscheidung aus (siehe 68000/68020 User's Manuals, Abschnitte Bus Operation). Zwei UND-Gatter in den 'Data Transfer and Size Acknowledge'-Leitungen $\overline{DSACK0}$ und $\overline{DSACK1}$ sorgen für entsprechende Eingänge am Erweiterungsstecker X2.

Aus den Steuersignalen des 68020 die des 68000 abzuleiten, ist zum größten Teil ähnlich simpel. Der 68020 gibt mit seinen Size-Signalen genügend Hilfestellung, um die 68000-Data-Strobes $\overline{LDS}$ und $\overline{UDS}$ leicht 'ergattern' zu können. Address Strobe und Bus Grant brauchen noch eine Inhibit-Bedingung, um die 68000-Hardware bei PAK-internen Aktivitäten (FPU- oder 32-Bit-RAM-Zyklen) abhängen zu können. Und schließlich machen drei Tri-State-Treiber die neuen Address- und Data-Strobe-Ausgänge DMA-fähig.





182 Beinchen unter 2 Pin-Grid-Aray-Gehäusen (PGA) — das ist 32-Bit-Anschlußtechnik.

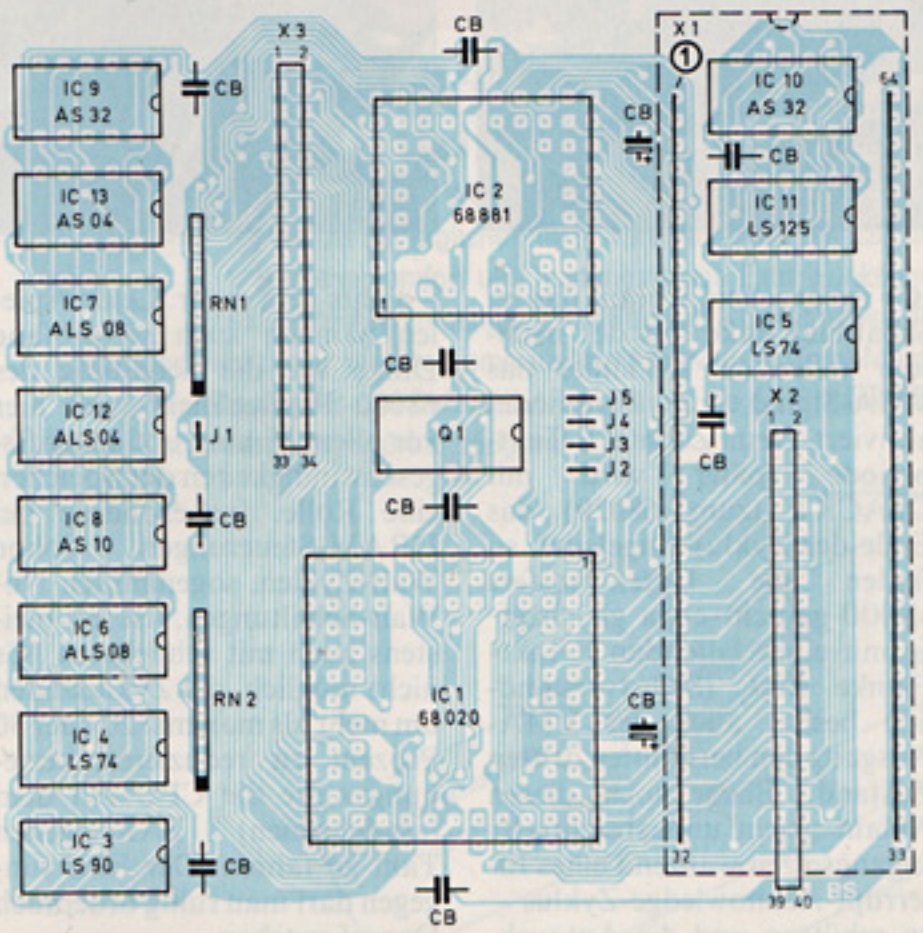


Stückliste

IC 1	68020	X 1	2 x 32pol. Stiftstreifen (Steckadapter zum 68000-Sockel)
IC 2	68881	X 2	40pol. Pfostenverbinder
IC 3	LS 90	X 3	34pol. Pfostenverbinder
IC 4,5	LS 74	J 1	1 x 2pol. Pfostenleiste
IC 6,7	ALS 08	J 2...5	2 x 4pol. Pfostenleiste
IC 8	AS 10		114pol. PGA-Fassung (CPU)
IC 9, 10	AS 32		68 pol. PGA-Fassung (FPU)
IC 11	LS 125		div. 14pol. DIL-Fassungen
IC 12	ALS 04		
IC 13	AS 04		
Q 1	Quarzoszillator für async. CPU-/FPU-Takt (typ. 12,5/16,667 MHz)		
RN 1,2	7 x 4 k 7 SIL		
CB	9 x 100 n keram. 2 x 10µ Tantal		

Das meiste Gehirnschmalz steckt eindeutig in der Nachbildung des synchronen Businterface, worüber der 68000 Anschluß an die alten 6800-Peripheriechips hatte; beim 68020 ist es nicht mehr eingebaut. Die

Da ein 68000-Sockel nicht verpolungssicher sein kann, ist unbedingt auf richtiges Einsetzen zu achten.

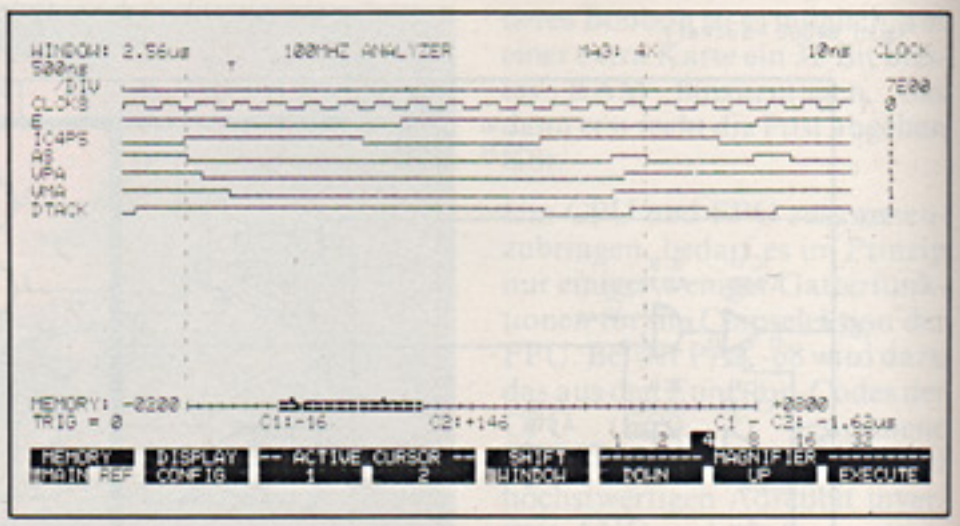


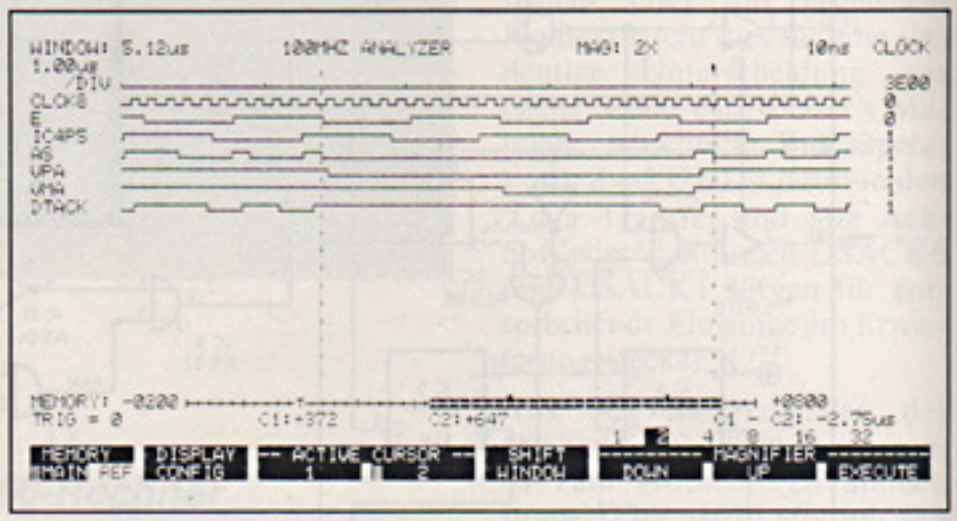
Pinbelegung X 2

Pin	Signal	Pin	Signal
1	$\overline{AS}$	21	A 16
2	GND	22	A 17
3	$\overline{CPUSPACE}$	23	A 14
4	A 5	24	A 15
5	SIZ 0	25	A 12
6	SIZ 1	26	A 13
7	A 0	27	A 10
8	A 30	28	A 11
9	A 28	29	A 1
10	A 29	30	A 9
11	A 27	31	A 2
12	A 26	32	A 7
13	A 25	33	A 3
14	A 24	34	A 8
15	A 21	35	A 4
16	A 23	36	A 6
17	A 20	37	VCC
18	A 22	38	VCC
19	A 18	39	$\overline{DSACK\ 1\ RAM}$
20	A 19	40	$\overline{DSACK\ 0\ RAM}$

Pinbelegung X 3

Pin	Signal	Pin	Signal
1	GND	18	D 8
2	GND	19	D 6
3	D 16	20	D 9
4	D 15	21	D 10
5	D 18	22	D 12
6	D 19	23	D 7
7	D 20	24	D 21
8	D 17	25	D 23
9	D 13	26	D 22
10	D 11	27	D 27
11	D 0	28	D 26
12	D 14	29	D 25
13	D 1	30	D 24
14	D 2	31	D 30
15	D 3	32	D 28
16	D 4	33	D 31
17	D 5	34	D 29





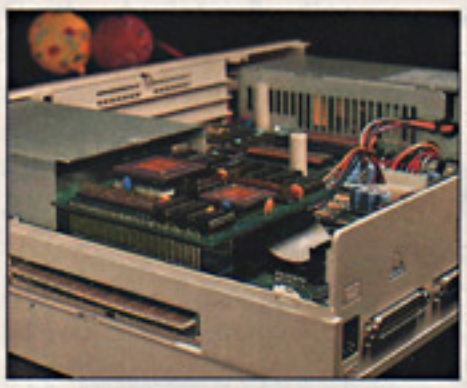
Der Logikanalysator bringt es an den Tag: Synchrone Zyklen kosten den 68020 mehr Takte als einen 68000 (oben: best case / unten: worst case).

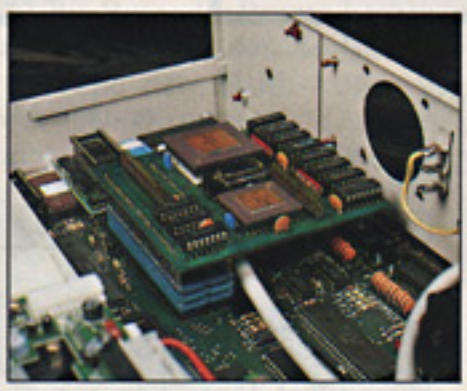
ganze Signalerzeugung dreht sich um den Frequenzteiler IC3, der wie beim 68000 den E-Takt asynchron aus dem Systemtakt 'erteilt' (Teilerfaktor 10). Abweichend vom Original erzeugt die Nachbildung ein symmetrisches E-Taktsignal; bislang sind jedoch noch keine Fälle bekannt geworden, in denen das gestört hätte.

Ein synchroner Buszyklus wird eingeleitet, wenn die adressierte (Port-)Schaltung mit $\overline{VPA}$ statt $\overline{DTACK}$ antwortet. Damit werden die beiden Flipflops von IC5 freigegeben, worauf das im Schaltbild linke mit der nächsten fallenden E-Flanke das $\overline{VMA}$-Signal (zur Peripherie) aktiviert. Neun Zehntel E-Taktperiode später wird mit $\overline{DSACK1}$ (zum 68020) das Ende des Zyklus eingeleitet; so finden die Datentransfers 68000-gerecht etwa gleichzeitig mit einer fallenden E-Takt-Flanke statt. (Selbstverständlich besitzt auch der $\overline{VMA}$-Ausgang den berühmten dritten Zustand.) Bliebe noch, diesen Ablauf gegen unbeabsichtigtes Ingangsetzen während eines Interrupt-Acknowledge-Zyklus zu schützen und die Autovektor-Option der CPU zu 'befreien' – beides geschieht mit dem internen Interrupt-Acknowledge-Signal am Ausgang eines 3fach-NAND-Gatters (IC8, Pin 8).

Nicht zu bremsen?

Für das Tempo, das Ihr Rechner mit der PAK-68 an den Tag legt, ist eigentlich nur die Taktfrequenz der Prozessoren ausschlaggebend. Also greift man zu den 16-MHz-Chips und dem passenden Quarzoszillator, und – nichts geht mehr. Leider spielen nämlich auch so profane Dinge wie die Schaltung des 68000-'Restrechners', und hier vor allem Timing und Zugriffsgeschwindigkeiten der Speicher, eine Rolle. Insbesondere bei DRAM-Steuerungen ohne Wait-Zyklen, sogenannten No-Wait-Schaltungen, ist es meistens auch mit schnelleren ICs nicht möglich, die Zykluszeiten um mehr als maximal 20 oder 30 Prozent zu reduzieren beziehungsweise die CPU mit dem (asynchronen) PAK-eigenen Takt zu fahren. Der FPU hingegen darf man ruhig ordentlich Dampf machen.





Drei Einsatzbeispiele für die PAK-68: Amiga 1000 (links; ein bißchen mechanische Nachhilfe ist vonnöten), Atari ST im separaten Gehäuse (Mitte) und c't 68 ECB. Ohne Software-Unterstützung nutzt die PAK allerdings nicht viel.

Je nach geplantem Einsatz der PAK ergeben sich also unterschiedliche Anforderungen an die Takterzeugung, denen mit den Steckbrücken J2. . .5 Rechnung getragen wird: J2(J4) holt den CPU-(FPU-)Takt vom 68000-Sockel, J3(J5) wählt den bordeigenen. Die jeweils optimale Kombination hängt wie üblich vom Einzelfall ab. Bei geschlossenen Systemen wie Ataris ST oder Commodores Amiga muß der CPU-Takt vom Motherboard bezogen werden. Bei VME-Bus-Platinen ist es zumeist nicht so kritisch, wo der Takt herkommt; dennoch gibt es bei 10–12 MHz so etwas wie eine magische Grenze, über die auch ein Speicherwechsel nicht unbedingt mehr hinweghilft. Immerhin ist das System dann allein mit der 68020-CPU um rund die Hälfte schneller als mit einem 8-MHz-68000.

Die universellste und zugleich billigste Grundkonfiguration dürfte daher die PAK-68 mit einer 12-MHz-CPU am 68000-Takt sein; natürlich tut es auch eine 16-MHz-Version. Den Kauf der FPU nebst Quarzoszillator kann man vertagen, bis wieder genügend Kleingeld im Portemonnaie ist. Bereits in dieser Mindestausbaustufe brachte die PAK in einem Atari ST unter RTOS-UH bis zu 40 Prozent mehr Geschwindigkeit. Dazu mußte lediglich der Cache des 68020 aktiviert werden.

Wo's (nicht) klappt

Leider zeigt sich nicht jedes Betriebssystem einem 68020 ge-

PAK-68 und ST

Atari-ST-Besitzer, die ihren Computer mit der geballten Rechenleistung der 68020/68881-Kombination ausstatten möchten, stehen an einem Scheideweg: TOS und GEM laufen auf dem 68020 nicht, und es ist noch nicht absehbar, ob wir die erforderlichen Änderungen entwickeln und veröffentlichen können. Wer 32-Bit-Power will, muß sich deshalb gleichzeitig für das Multitasking-Betriebssystem RTOS-UH entscheiden, das sich mit allen Prozessoren der 68000-Familie versteht.

Unser Tip: Wer einen 68020-ST haben und auf GEM/TOS nicht verzichten möchte, sollte sich eine zweite Rechnerkonsole (ohne Peripherie) kaufen und diese umrüsten. Unter dem Einfluß der Mega-ST-Euphorie werden gebrauchte Geräte gelegentlich schon zu Schleuderpreisen angeboten.

Auf dem mittels PAK-68 aufgebohrten Atari ST läuft die aktuelle RTOS-Version 2.0 in den beiden EPROM-residenten Varianten A und B problemlos. Besitzer der Disketten-Version (C) haben aber noch eine Hürde zu überwinden, bevor sie das Betriebssystem von der Diskette booten können: Auch der TOS-Urlader enthält unverträgliche Zeitschleifen, so daß der Boot-Vorgang unter 68020-Regie nicht funktioniert. Wir hoffen, in der nächsten c't-Ausgabe eine Patch-Anleitung zur Beseitigung dieses Problems abdrucken zu können.

Der flotte Zahlenknacker 68881 kann derzeit auch unter RTOS-UH/PEARL noch nicht voll ausgefahren werden. In einem Multitasking-Betriebssystem müssen nämlich bei jedem Task-Wechsel auch die verwendeten Register des Coprozessors umgeladen werden, damit es keinen Zahlensalat gibt. Assembler-Programmierer könnten zwar die Rechenleistung der FPU gelegentlich schon nutzen – allerdings nur im 'Dispatcher-Off', also unter Blockierung des Task-Wechsels.

Es wird deshalb eine neue Betriebssystem-Version vorbereitet, bei der die Vorkehrungen für den FPU-Einsatz getroffen sind. Darüber hinaus werden ein PEARL-Compiler mit voller FPU-Unterstützung sowie ein 68020-Assembler angeboten. Betriebssystem-Update und Zusatzpaket werden voraussichtlich ab Oktober erhältlich sein. Auch der in Entwicklung befindliche Fortran-77-Compiler für RTOS-UH (verfügbar voraussichtlich ab Dezember) wird den Einsatz des Coprozessors optional unterstützen.

genüber so verständnisvoll wie RTOS-UH. So wird es die Atari-ST-User betrüben, daß ihr GEM/TOS aktuell noch nicht mit der PAK-68 klarkommt. Überhaupt sind alle Betriebssysteme ausgeschlossen, die nach einem Bus- oder Address-Error unmittelbar auf die 'gestackten' Registerinhalte zugreifen – die sind beim 68020 nun einmal anders als beim 68000. (Diese Systeme laufen übrigens auch nur selten mit einem 68010).

Dagegen haben Systeme, deren 68000 man durch einen 'Zehner' ersetzen kann, in der Regel keine weiteren Softwareprobleme; Commodores Amiga zum Beispiel lief mit der PAK auf Anhieb – wenigstens auf der Betriebssystem-Ebene. Inwieweit nämlich die Anwender-Software genauso 'wasserdicht' programmiert ist, steht auf einem ganz anderen Blatt...

Auch Programme, die ein kribbeliges Timing mit Zeitschleifen erzeugen, arbeiten oft nicht mehr zuverlässig, weil der 68020 in seinem Cache einfach zu schnell ist. Um ein nicht funktionierendes Programm dahingehend zu prüfen, kann man mit Jumper J1 den Cache-Speicher außer Betrieb setzen (J1 gesteckt); der Geschwindigkeitsvorteil beträgt dann im Schnitt unter 10 Prozent. Läuft das Programm jetzt immer noch nicht, liegt wahrscheinlich eine Stack-Manipulation oder unsaubere Adressierung vor.

Es wird eng

Unvermutet kann sich noch ein weiteres nicht zu widerlegendes Argument gegen die PAK-68 ergeben: die Mechanik. Die Platine braucht ihre 10 × 10 × 2 cm Luft und den 68000-Sockel in der richtigen Richtung. Wir hoffen, mit der gewählten Anordnung nicht allzu sehr danebenzuliegen. Die Bilder zeigen, wie der Einsatz der PAK aussehen kann – im Sinne des Wortes. Daß es mitunter nicht ohne Säge und 'Feilchen' geht, mag zwar ärgerlich sein, ist aber immer noch besser, als wenn's gar nicht paßt.

Eine fast unmögliche Situation entsteht allerdings ausgerechnet beim c't68000, wo die PAK die Ebene der Busplatine kreuzt; damit sie paßt, muß schon einiges an gehäusetechnischen Zufällen zusammenkommen. Dennoch erschien es uns vertretbar, auf diesen speziellen Einsatzfall zu verzichten – bald wird es nämlich auch eine c't68020-CPU-Karte mit 16- und 32-Bit-Bus und 32 Bit breitem EPROM geben, die doch ein bißchen mehr 68020 'rüberbringt als eine Austauschlösung für einen 68000-Sockel.

```
 1  *-----------------------------------------------*
 2  * cache=2.1                              15.06.87 *
 3  *-----------------------------------------------*
 4  * (c) 1987 esd schulze & detering hannover brd   *
 5  *-----------------------------------------------*
 6  * additional cache link for PAK-68               *
 7  *                                                *
 8  * activation by LOAD plus ABORT                  *
 9  *-----------------------------------------------*
10  *                          Task header:          *
11          DC.L    0,0        RTOS internal link    *
12          DC.W    $41        autostart task        *
13          DC.B    '#cache'   started once&never again*
14          DC.W    14         #cache priority       *
15          DC.L    $120       #cache task workspace *
16          DC.L    0,0        #cache system link    *
17          DC.W    14         #cache priority       *
18          DC.L    START      #cache start pc       *
19          DC.L    0,0,0,0,0,0,0,0,0,0              *
20          DC.L    0,0,0,0,0,0,0,0,0,0              *
21  *-----------------------------------------------*
22     TERMI:  OPD  $4E41  RTOS TRAP terminate task  *
23     OFF:    OPD  $4E4F  RTOS TRAP supervisor state*
24     ILVECT: EQU  $010   ILLEGAL INSTR. VECTOR     *
25     VCALNK: EQU  $4B8   CACHE enable link (TRAP)  *
26     DPC:    EQU  $800   DISPATCHER awake flag     *
27     DISEX:  EQU  $906   DISPATCHER awake entry    *
28  *-----------------------------------------------*
29  START  LEA      $120(A4),A7  cache link entry    *
30         OFF                  supervisor state     *
31         MOVE.L   A7,D1        save SV-stackpointer*
32         MOVE.L   ILVECT,D2     and ILLEGAL vector *
33         LEA      ILLEG,A0     new ILLEGAL entry    *
34         MOVE.L   A0,ILVECT    setup vector         *
35         MOVEQ    =1,D0        'enable cache' code *
36         DC       $4E7B,2      'MOVEC D0,CACHCR '   *
37  *                            skip if 68000/10     *
38  *                            else supply (new)    *
39         MOVE.L   CACL,VCALNK   cache clear trap    *
40  ILLEG: MOVE.L   D2,ILVECT    reload ILLEGAL vect.*
41         MOVEA.L  D1,A7         and SV-stackpointer*
42         ANDI     =$D8FF,SR    back to user state   *
43         TERMI                 all done             *
44  *-----------------------------------------------*
45  *   exception link for cache clear trap ($A05C): *
46  CACL:  MOVEQ    =9,D0        'cache clear' code   *
47         DC       $4E7B,2      'MOVEC D0,CACHCR '   *
48  *                            RTOS exception exit:*
49         ORI      =$700,SR     IRQ off mask         *
50         TST      DPC          test: dispatcher     *
51         BMI.S    CACLX        b:    called         *
52         RTE                   else: quick exit     *
53  CACLX  JMP      DISEX        awake dispatcher     *
54  *-----------------------------------------------*
55         END
```

Ein paar zig Zeilen Assembler-Programm machen RTOS-UH 68020-fähig.