

CEPAC-65:

Der steuert (fast) alles

Christian Persson

Software anstelle von festverdrahteter Logik — dieses Konzept liegt unserer CEPAC-Serie zugrunde. Mit dem CMOS-Einplatinen-Allzweck-Computer CEPAC-80 hat c't (in Heft 1/84) bereits einen recht komfortabel ausgestatteten Kleinstcomputer für diesen Anwendungsbereich vorgestellt. Jetzt folgt der CEPAC-65 — ein Minimalsystem, das alle Chancen hat, Deutschlands preisgünstigster Steuerungscomputer zu werden.

Das 'Rechenzentrum' des CEPAC-65 ist die CPU 6502 — bekanntlich der weltweit am meisten verwendete Mikroprozessor. Die Programmentwicklung für den CEPAC-65 kann auf jedem Computer erfolgen, der mit dieser CPU ausgestattet ist, also beispielsweise auf dem AIM-65, einem Apple- oder Commodore-Rechner. Selbstverständlich kann man auch ein Z80-System mit Cross-Assembler verwenden. Besonders gut eignet sich der in unserer Schwesterzeitschrift elrad vorgestellte COBOLD-Computer, der auch im Hinblick auf die Hardware weitgehend mit dem CEPAC-65 kompatibel ist.

Das 'C' in CEPAC-65 weist darauf hin, daß der Computer vollständig mit CMOS-Bausteilen

bestückt werden kann. Der Strombedarf der gesamten Einheit liegt in dieser Version bei nur rund 12 mA (je nach EPROM-Typ). Ein weiterer Vorzug der CMOS-Version ist die höhere Störspannungsfestigkeit. Darüber hinaus besitzt die CMOS-CPU einen erweiterten und somit leistungsfähigeren Befehlssatz.

Die CMOS-Version der 6502-CPU wird von mehreren Firmen hergestellt — beispielsweise von Rockwell (Typenbezeichnung: R65C02) und von GTE Microcircuits (Typenbezeichnung: G65SC02). Die Rockwell-CPU besitzt einen um vier Befehle größeren Befehlssatz. Der RIOT-Baustein wird nach unseren Informationen zur Zeit nur von GTE in einer CMOS-Ausführung angeboten (Typenbezeichnung: G65SC32).

Die NMOS-Ausführung unseres Kleinstcomputers stellt etwas höhere Anforderungen an die Stromversorgung (Strombedarf: rund 250 mA). Wo das keine Rolle spielt, kann man aus dem 'CEPAC' einen besonders preisgünstigen 'EPAC' machen. Nach unserer überschlüssigen Berechnung läßt sich die NMOS-Version für weniger als 80 Mark aufbauen.

Viel Platz für Festprogramme

In seinem typischen Anwendungsbereich arbeitet der CEPAC-65 mit einem Festprogramm, das sich im EPROM befindet. Ein Arbeitsspeicher (RAM) wird natürlich ebenfalls benötigt — beispielsweise für den Stack, auf dem der Prozessor beim Sprung in Unterprogramme die Rückkehradressen ablegt, und zur Speicherung von Variablen. Allerdings braucht der Schreib-/Lesespeicher nicht besonders groß zu sein, wogegen das Festprogramm einen beträchtlichen Umfang haben kann.

Nach diesen Kriterien richtet sich die Speicherausstattung des CEPAC-65: Als Arbeitsspeicher stehen 128 Bytes statisches RAM im RIOT 6532 zur Verfügung, die bei geschickter Programmierung für die meisten Anwendungen genügend Platz bieten. Die Kapazität des EPROM-Bereichs kann nahezu jedem Bedarf angepaßt werden — bei Verwendung des EPROM-Typs 27256 lassen sich stolze 32 KByte unterbringen!

Wer mit dem auf der Karte zur Verfügung stehenden Speicherplatz nicht auskommt oder gar den CEPAC-65 zu einem größeren System ausbauen möchte, sei auf die optionale Ausstattung verwiesen: An der Stirnseite der Platine kann man eine 64-polige Messerleiste anbringen, an der dann alle wichtigen Prozessorleitungen zur Verfügung stehen. Eine zusätzliche Memory-Deselect-Leitung erlaubt es, extern eine weitergehende Adreßdekodierung vorzunehmen und den auf der Karte vorhandenen Speicher bei Bedarf abzuschalten.

An der Messerleiste liegen übrigens auch alle 16 Portleitungen, die der CEPAC-65 zur Aufnahme von Daten oder zur Steuerung angeschlossener Baugruppen zur Verfügung stellt. Die Ein-/Ausgabe erfolgt über den bereits erwähnten RIOT-Baustein (RAM, In/Out, Timer) 6532, der zwei 8-Bit-Ports, einen programmierbaren Interrupt-Timer und einen Flankendetektor besitzt. Der Timer erweist sich als besonders nützlich beim Einsatz für Steuerungszwecke. Wie der RIOT-Chip programmiert wird, erläutern wir in dieser

Ausgabe unter der Rubrik 'c't-Applikation'.

Für den CEPAC-65 haben wir zwei Platinenversionen vorgesehen: Version A hat die Abmessungen 100 x 80 mm (halbes Europaformat) und ermöglicht besonders platzsparende Anwendungen. Ein Lochrasterstreifen von 38 x 4 Lötstellen erlaubt es dennoch, Steckverbinder und sogar Treibertransistoren oder ähnliche Bauelemente zusätzlich unterzubringen. Version B besitzt Europaformat und bietet mit einem 38 x 34-Lochrasterfeld sehr viel Platz für frei verdrahtete Anwendungsschaltungen. Alle Portleitungen sind an die erste Lochreihe geführt.

Adressenlage

Eine vollständige Adreßdekodierung ist bei einem Minimalsystem von der Art des CEPAC-65 nicht nur überflüssig — sie kann sich sogar als nachteilig erweisen: Die wegen der unvollständigen Dekodierung auftretende 'Spiegelung' des EPROM-Bereichs — ein 2-KByte-EPROM beispielsweise scheint 16-fach vorhanden

X1		
a		c
Masse	1	Masse
RES in	2	RES out
Φ0	3	MEM Deselect
Φ2	4	RIW
PA1	5	PA0
PA3	6	PA2
PA5	7	PA4
PA7	8	PA6
PB1	9	PB0
PB3	10	PB2
PB5	11	PB4
PB7	12	PB6
NMI	13	SYNC
IRQ	14	ML*
RDY	15	Φ1
A1	16	A0
A3	17	A2
A5	18	A4
A7	19	A6
A9	20	A8
A11	21	A10
A13	22	A12
A15	23	A14
BE*	24	s. o.
D1	25	D0
D3	26	D2
D5	27	D4
D7	28	D6
b	29	a
d	30	c
f	31	e
U	32	5 V

* bei bestimmten 65xx-Versionen (z. B. 65SC102, 65SC112)

Tabelle 1. Signale an der 64-poligen Messerleiste, die bei Bedarf eingesetzt wird.

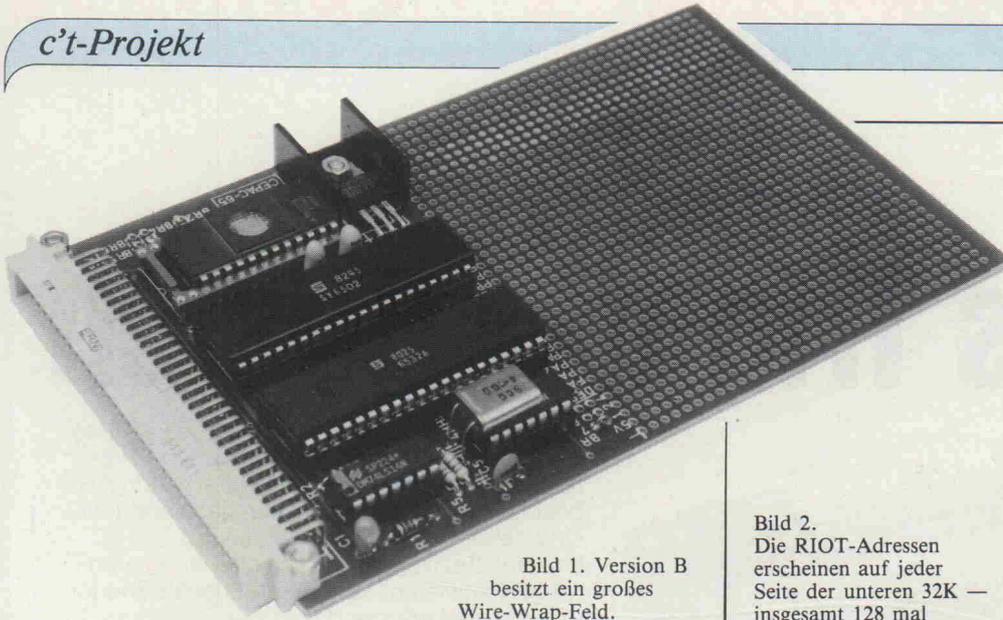


Bild 1. Version B besitzt ein großes Wire-Wrap-Feld.

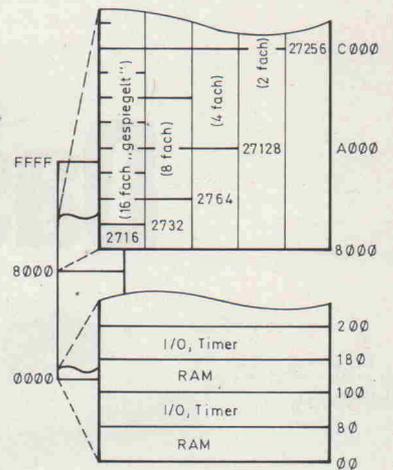


Bild 2. Die RIOT-Adressen erscheinen auf jeder Seite der unteren 32K — insgesamt 128 mal

zu sein — kann die Programmentwicklung beträchtlich erleichtern. Der Programmierer kann das Programm in einem Adreßbereich seines Computers entwickeln und testen, der mit freiem Arbeitsspeicher belegt ist. Auch die vielfach 'gespiegelten' Register des RIOT las-

sen sich auf diese Weise leichter ansprechen. Bild 2 zeigt, daß auch das RAM unter vielen Adressen erreicht werden kann. RAM und RIOT-Register (inklusive der Ports) belegen eine Speicher-'Seite' (256 Byte). Da zur Bildung des Select-Signals für den RIOT-

Baustein lediglich das Most Significant Bit (MSB) der Adresse verwendet wird, erscheint der RIOT auf jeder 'Page' in der gesamten unteren Hälfte des Adreßbereichs — also ganze 128mal. Übrigens wäre ohne eine solche Mehrfachadressierung der Be-

trieb des 6502-Prozessors in einem System mit nur 128 Byte RAM kaum möglich: Kenner der 65er-Familie wissen, daß die CPU einerseits die Speicherseite 01 für den Stack benötigt, und daß andererseits ohne die Seite 00 ('Zero Page') die leistungsfähigsten Adressie-

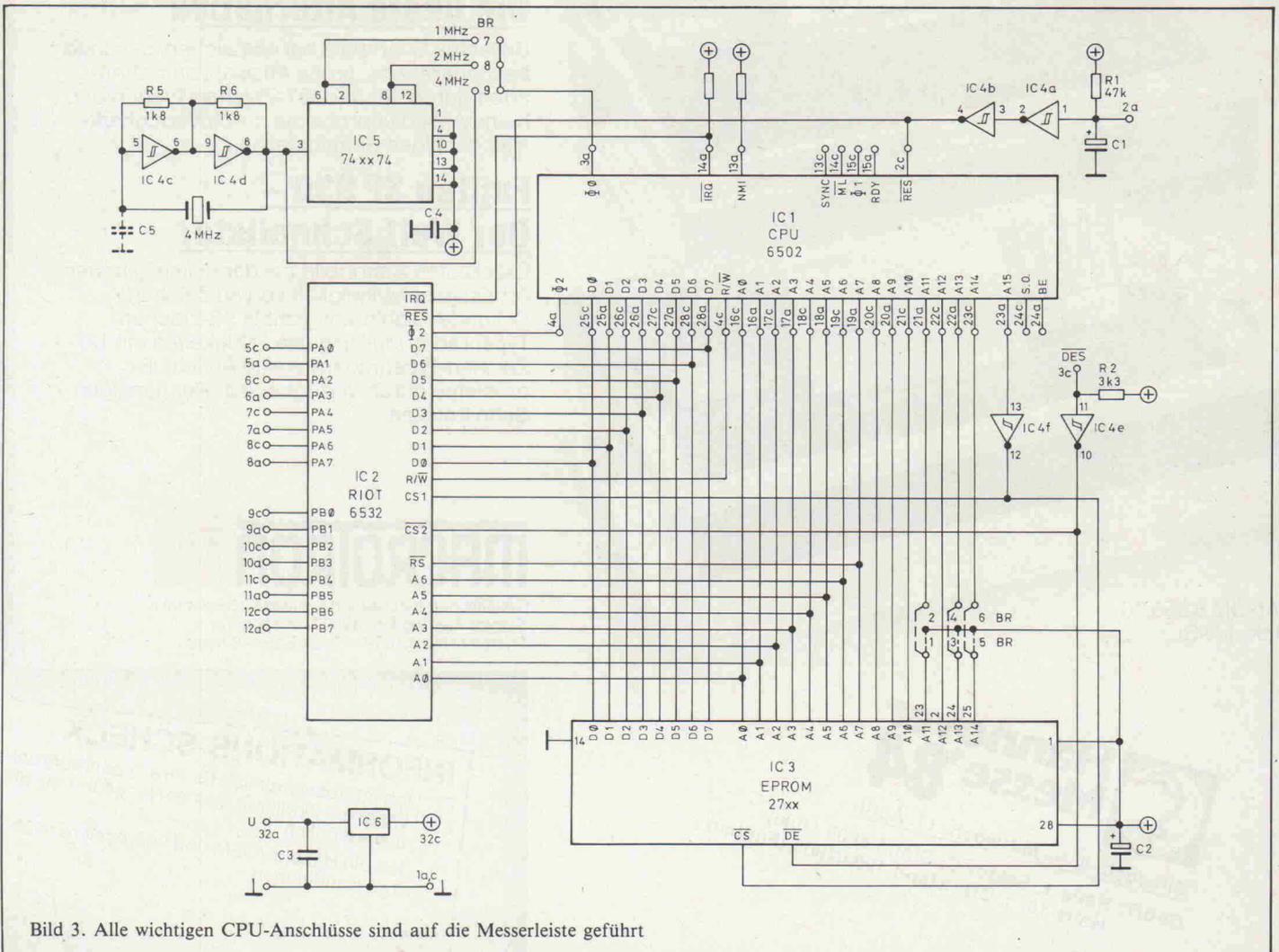


Bild 3. Alle wichtigen CPU-Anschlüsse sind auf die Messerleiste geführt

rungsarten nicht verwendet werden könnten.

Bei der Entwicklung von CEPAC-65-Programmen ist zu beachten, daß der Arbeitsspeicher in der unteren Hälfte jeder Speicherseite liegt. Der Stackpointer muß infolgedessen mit 7Fh initialisiert werden, und nicht, wie gewöhnlich, mit FFh.

Klar, daß Variablen nur den unteren Teil des RAM belegen dürfen. Der obere Teil ist für den Stack reserviert, der von 7Fh abwärts 'wächst'. Während in 'großen' 65er-Systemen bis zu 256 Byte als Stack zur Verfügung stehen, muß sich die CPU beim CEPAC-65 den knappen Speicherplatz mit dem Programmierer teilen. Man sollte deshalb — vor allem beim Entwurf wohlstrukturierter Programme — darauf achten, daß Variablenbereich und Stack nicht miteinander in Konflikt geraten, weil zuviele Unterprogrammaufrufe ineinander verschachtelt sind.

Aufbau

Der Aufbau des CEPAC-65 auf der durchkontaktierten und mit Bestückungsaufdruck versehenen Platine stellt keine besonderen Anforderungen. Selbstverständlich sollten Sie beim Umgang mit CMOS-Bau-elementen die üblichen Vorsichtsmaßnahmen gegen statische Elektrizität treffen. Achten Sie bitte auch auf die Polung der ICs und Tantal-Kondensatoren. Trotz Lötstopplack sind Lötfehler bei der relativ engen Leiterbahnführung

nicht auszuschließen. Gehen Sie deshalb beim Einlöten der Bauteile sehr sorgfältig vor. Wie 24-polige EPROMs in die 28-polige Fassung eingesetzt werden müssen, zeigt der Bestückungsplan.

Tabelle 2 können Sie entnehmen, welche Brücken für welche EPROM-Typen eingesetzt werden müssen. Eine weitere Brücke in einem der Felder 7...9 legt die System-Taktfrequenz fest. Normalerweise trägt diese 1 MHz (Brücke 7);

Stückliste

Widerstände
R1 ~~3k3~~ 47k
R2...4 ~~47k~~ 3k3
R5,6 1k8

Kondensatoren
C1,2 4µ7, min. 10 V,
Tantal
C4 100nF
(C5 15pF⁽¹⁾*)

Halbleiter
IC1 R6502⁽¹⁾, G65SC02⁽²⁾
IC2 R6532⁽¹⁾, G65SC32⁽²⁾
IC3 EPROM nach Bedarf
(s. Tabelle 2), 450 ns,
ggfs. CMOS-Typ
IC4 74LS14⁽¹⁾, 74HC04⁽²⁾
IC5 74LS74⁽¹⁾, 74HC74⁽²⁾

Verschiedenes
Platine Version A
(80 x 100 mm)
bzw.
Version B
(160 x 100 mm)

Optional
IC6 7805⁽¹⁾, 78L05⁽²⁾
C3 1µF, 35 V, Tantal

⁽¹⁾ NMOS-Version
⁽²⁾ CMOS-Version
* siehe Text

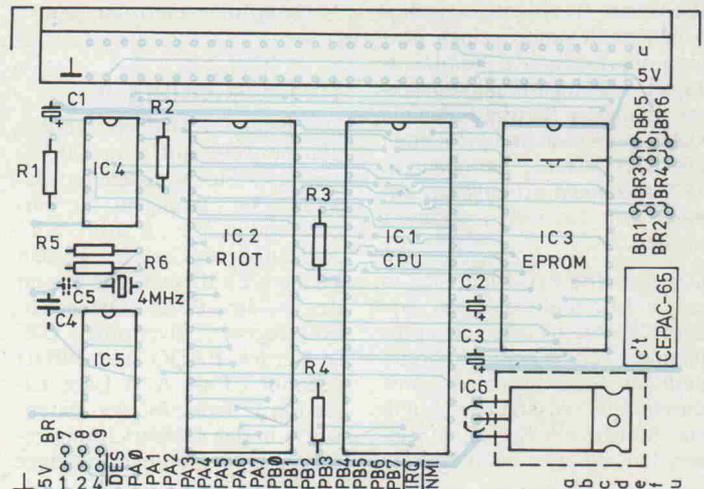
Tabelle 2

EPROM	Brücken
2716	1, 3
2732	2, 3
2764	2, 3, 5
27128	2, 4, 5
27256	2, 4, 6

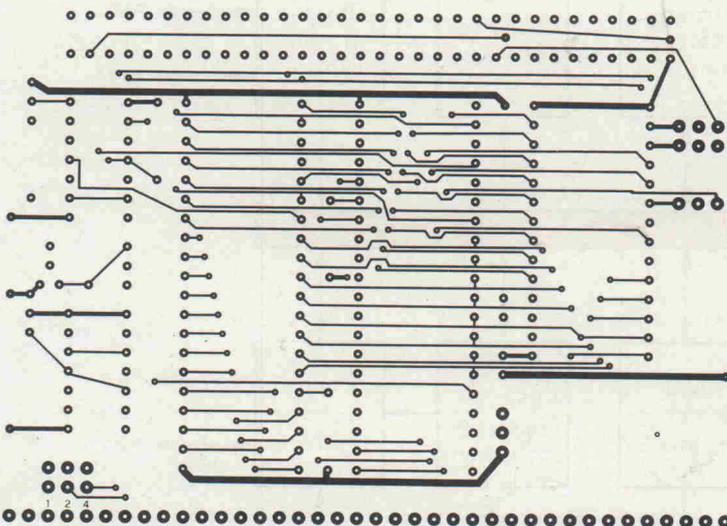
nur, wenn Sie schnellere Versionen der CPU, des RIOT und des EPROMs verwenden, darf ein schnellerer Systemtakt eingestellt werden. Beim Aufbau des Oszillators unter Verwendung von Invertiern mit Schmitt-Trigger-Eingängen (IC4) kommt es in Einzelfällen vor, daß der Quarz auf einer Oberwelle schwingt. Ein kleiner Kondensator (C5) löst dieses Problem. Er sollte nicht eingesetzt werden, wenn der Takt-generator einwandfrei arbeitet.

Die Spannungsversorgung mit einer geregelten und stabilisier-

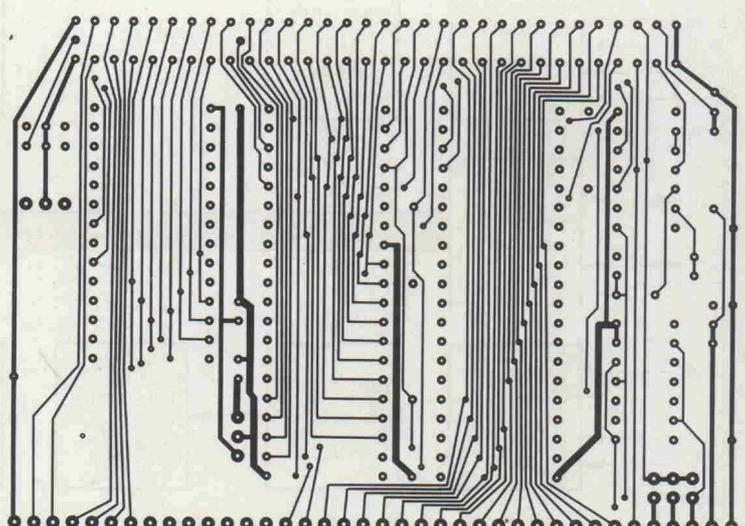
ten Gleichspannung kann an den im Bestückungsplan mit 5V gekennzeichneten Anschlüssen erfolgen. Bei Bedarf läßt sich auch ein Spannungsregler auf der Platine unterbringen. Die Versorgungsspannung wird dann an einem der mit U gekennzeichneten Punkte zugeführt. Sie sollte zwischen 7,5 und 12 Volt liegen. (Die Spannungsregler verkräften zwar bis zu 30 Volt am Eingang, die steigende Verlustleistung würde aber zusätzliche Kühlmaßnahmen erfordern.) Bitte beachten Sie, daß bei Einsatz eines Spannungsreglers auch der zusätzliche Kondensator C3 eingesetzt werden muß. Eine auf dem Lochrasterfeld eventuell zusätzlich aufgebaute Schaltung kann aus der geregelten Spannung versorgt werden, soweit das nicht die Belastbarkeit des Spannungsreglers überschreitet.



Die Bestückung erfolgt wahlweise mit NMOS- und TTL- oder mit CMOS-Bausteinen.



Bestückungsseite



Lötseite

RIOT 6532: Einsatz und Programmierung

Christian Persson

Jeder Computer benötigt Ein-/Ausgabeleitungen, um mit der Umwelt in Datenverkehr treten zu können, und er braucht einen Arbeitsspeicher, um beispielsweise Variablen und Rücksprungadressen aufbewahren zu können. Bei unserem Minimal-Computer CEPAC-65, der in dieser Ausgabe vorgestellt wird, übernimmt ein IC beide Funktionen: Der RIOT-Baustein 6532 stellt sowohl zwei 8-Bit-Ports als auch 128 Bytes statisches RAM zur Verfügung. Außerdem enthält das IC einen programmierbaren Countdown-Timer und einen Flankendetektor. In diesem Beitrag sollen die vielen Einsatzmöglichkeiten des RIOT-Chips, den man in 6502-Systemen oft einsetzt, anhand von Beispielen erläutert werden.

Das statische RAM läßt sich in einem Minimal-Computer wie dem CEPAC-65 besonders problemlos verwenden: Während man in 6502-Systemen sonst durch eine zusätzliche Logik ein RAM-R/W-Signal zu bilden hat, ist dies beim Einsatz

des RIOT nicht erforderlich. Die CPUs der 65er-Familie verwenden den Datenbus nur in der zweiten Hälfte eines jeden Taktzyklus. Das vom Prozessor gebildete Taktsignal $\Phi 2$ zeigt an, daß die Daten auf dem Bus gültig sind. Zur Bildung des Signals RAM-R/W werden gewöhnlich $\Phi 2$ und R/W durch Gatter verknüpft. Beim Einsatz des 6532 kann man auf die Gatterschaltung verzichten, weil der Chip mit der Prozessorleitung $\Phi 2$ verbunden ist und die Verknüpfung intern ausführt.

16 I/O-Leitungen

Die beiden I/O-Ports werden üblicherweise mit A und B bezeichnet. Sie bestehen aus jeweils acht Leitungen, die einzeln wahlweise als Ein- oder Ausgänge 'deklariert' werden können. Zu diesem Zweck sind den beiden Ports 'Datenrichtungsregister' zugeordnet. Diese werden PADD und PBDD genannt ('Port A/B Data Direction'). Jedes Bit der Datenworte in den Datenrichtungsregistern korrespondiert mit einer

Portleitung (Bild 1): Eine '0' macht die Portleitung zu einem Eingang, eine '1' deklariert den Anschluß als Ausgang. Eine bestimmte Konfiguration von Ein-/Ausgabeleitungen läßt sich also sehr einfach dadurch herstellen, daß man die entsprechenden Daten in die Richtungsregister schreibt.

Die in Bild 2 dargestellte Konfiguration ließe sich also durch folgende Befehlssequenz einstellen:

```
LDA #3F
STA PADD
```

Den beiden Ports sind außerdem die Datenregister PAD und PBD ('Port A/B Data') zugeordnet. Diese beiden Register dienen als Zwischenspeicher (Latch) für Daten, die an den Ports ausgegeben werden sollen. Portleitungen, die als Eingänge deklariert sind, werden durch die interne Logik von den Datenregistern getrennt.

Die CPU kann in die Datenrichtungsregister und in die Datenregister schreiben, als handele es sich um gewöhnliche

RAM-Zellen. (Bekanntlich sind in 6502-Systemen die I/O-Ports 'memory-mapped'; sie liegen in demselben Adreßraum wie der Speicher.) Um alle als Ausgänge deklarierten Leitungen von Port A auf log. 0 zu legen, könnte man zum Beispiel diese Befehlsfolge ausführen lassen:

```
LDA #00
STA PAD
```

Als Ergebnis wird sich an den Portschlüssen PA0...PA5 der log.-0-Pegel einstellen, wenn der Port entsprechend Bild 2 konfiguriert ist. Der Pegel an den als Eingänge deklarierten Leitungen PA6 und PA7 wird natürlich nicht beeinflusst, obwohl auch die Bits 6 und 7 im Datenregister PAD auf 0 gesetzt worden sind.

Nehmen wir an, die beiden höchstwertigen Anschlüsse würden von außen auf log. 1 gelegt (offene Eingänge liegen ebenfalls auf 1). Liest die CPU nun den Inhalt des Registers PAD (zum Beispiel mit LDA PAD), so übernimmt sie das Datenwort 11000000, entsprechend C0h. Die Bits 0...5 entsprechen denen im Datenregi-

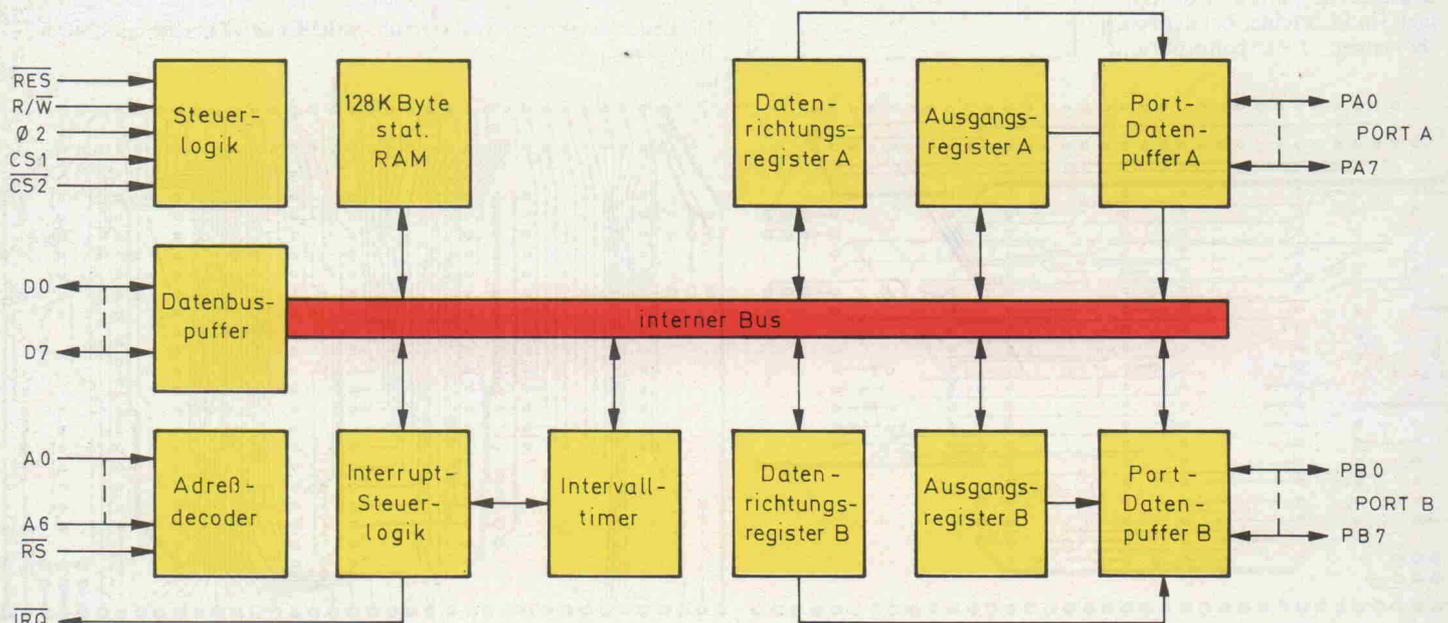
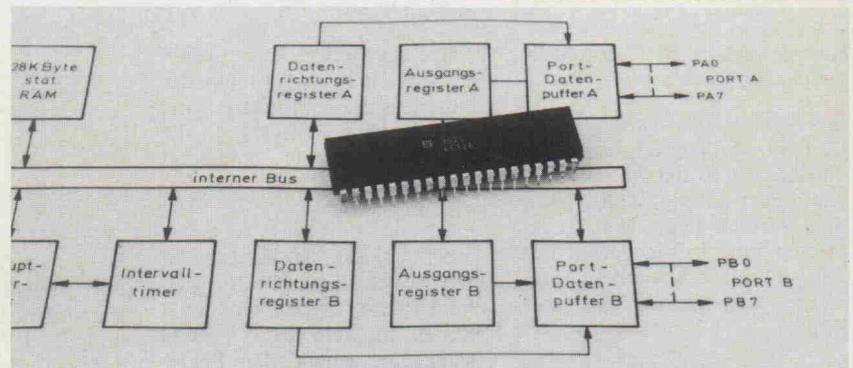


Bild 1. Innere Struktur der 6532

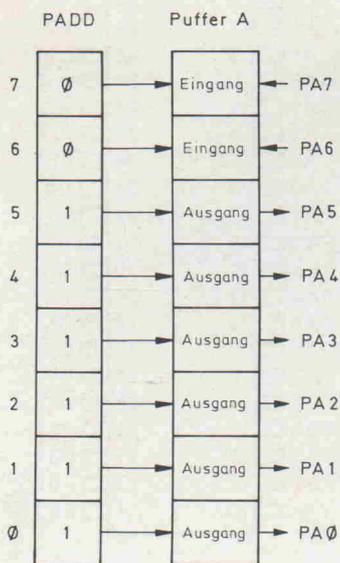


Bild 2. Deklarieren von Ein- und Ausgängen durch Datenrichtungsregister

ster, die Bits 6 und 7 repräsentieren den Pegel an den beiden als Eingänge definierten Leitungen. Beim Lesevorgang werden die als Eingänge deklarierten Leitungen direkt auf den Datenbus geschaltet, die betreffenden Bitwerte geben wieder den momentanen Pegel des anliegenden Signals.

Elektrisch verhalten sich die Portleitungen ähnlich wie LS-TTL-Anschlüsse: Pegel zwischen 0 und 0,8 Volt werden als log. 0 erkannt, Pegel zwischen 2 und 5 Volt gelten als log. 1; der Bereich dazwischen ist 'nicht erlaubt'. Ein als Eingang geschalteter Portanschluß stellt eine LS-TTL-Last dar. Die als Ausgang verwendete Portleitung kann eine LS-TTL-Last treiben. Port B besitzt im Unterschied zu Port A spezielle Treiber an den Ausgängen und kann einen Quellstrom von 3 mA bei 1,5 V liefern; er ist damit zum direkten Anschluß von Darlington-Leistungstreibern

geeignet. Die Ausgangsbelastung wirkt übrigens auf das Datenregister zurück: Bei einem Lesevorgang kann der Pegel nicht korrekt erkannt werden, wenn die Spannung am Ausgang infolge Belastung im Bereich zwischen 0,4 und 2,4 Volt liegt.

Das folgende kleine Programm erzeugt ein Rechteck-Signal an einer der Portleitungen. Dies geschieht, indem der Pegel in regelmäßigen Abständen zwischen 0 und 1 umgeschaltet wird. Die Variable TIME bestimmt die Dauer der Wartezeit zwischen den Schaltvorgängen und damit die Frequenz. Vorausgesetzt wird bei diesem Programm, daß die betreffende Portleitung zuvor als Ausgang deklariert worden ist.

```

SQUARE: LDA PBD
        EOR #20          ;Bit 5 invertieren
        STA PBD
        LDX TIME
        DEX
LOOP:   BNE LOOP        ;Zeit abwarten
        BEQ SQUARE      ;wieder invertieren

```

Die drei Instruktionen zu Beginn zeigen, wie man es bewerkstelligt, daß sich der Pegel an nur einer Ausgangsleitung ändert, wenn der an den übrigen nicht bekannt ist. Mit dieser Instruktionsfolge sollte man immer dann arbeiten, wenn an dem betreffenden Port mehrere Geräte angeschlossen sein könnten.

Hier ein simples Programm zur Überwachung des Pegels an einem Eingang:

```

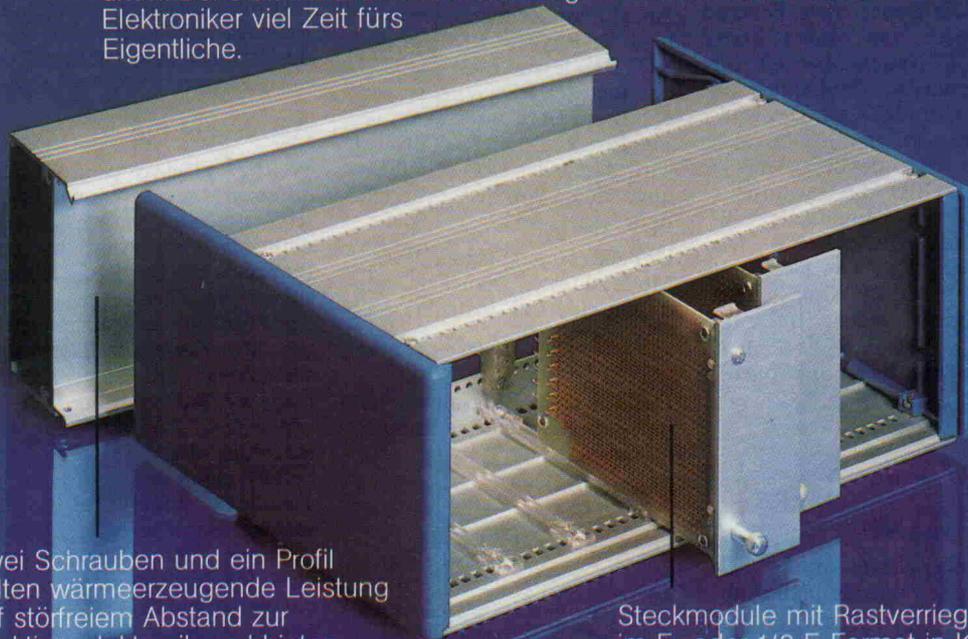
POLLIN: LDA PAD
        AND #80          ;Bits 0...6 maskieren
        BNE POLLIN      ;Bit 7 nicht 0: wiederholen

```

Es wird das sogenannte Polling-Verfahren angewandt: Die CPU fragt in einer Programmschleife so lange den Eingang ab, bis sich der betreffende Bitwert von 1 auf 0 ändert. Der BIT-Befehl des 6502 erlaubt eine Vereinfachung des Programms, wenn es sich um eine der höchstwertigen Portleitungen handelt. Man verwendet deshalb bevorzugt diese

DER KMT ODER DAS KMT?

Beides: **Der Klein-Modul-Träger** ist ein zweiteiliges Gehäusemagazin mit einem neuen Konzept – Funktion vorne, Wärme hinten. **Das KMT-System**, für kompakte Elektronik mit kleinem Volumen in Stecktechnik, umfaßt Frontplatten, Karten, Stecker, Busplatten, Netzteile u.v.m. Bei erstaunlichem Preis-Leistungs-Verhältnis bietet es dem Elektroniker viel Zeit fürs Eigentliche.



Zwei Schrauben und ein Profil halten wärmeerzeugende Leistung auf störfreiem Abstand zur Funktionselektronik und bieten besten Zugang zur Verdrahtungsebene.

Steckmodule mit Rastverriegelung im E- oder 1/2 E-Format in Karten oder Chassis-Ausführung.

BICC vero KMT erhalten Sie im Fachhandel zum vernünftigen Preis



Leitungen für die Abfrage von Statusanzeigen bei Peripheriegeräten:

```
POLLIN: BIT PAD
        BMI POLLIN ;Warten,
                bis Bit 7 = 0
```

(für die Überwachung von PA7)

```
POLLIN: BIT PAD
        BVS POLLIN ;Warten,
                bis Bit = 0
```

(für die Überwachung von PA6)

Flankendetektor

Tritt an der überwachten Leitung ein sehr kurzer Impuls auf, so kann es allerdings vorkommen, daß der CPU dies entgeht, weil zum Zeitpunkt der Abfrage schon der vorherige Pegel wiederhergestellt ist. Der RIOT-Baustein bietet jedoch mit dem Flankendetektor eine Möglichkeit an, dieses Problem auszuschalten. Der Detektor erkennt — je nach Programmierung — eine negative oder positive Flanke an der Leitung PA7. Dies wird durch Setzen einer Flag in einem Statusregister des RIOT-Chips angezeigt. Das Register heißt RDFLAG; die sogenannte PA7-Flag ist das Bit 6. Sein Wert kann im Polling-Verfahren durch eine BIT-Instruktion abgefragt werden; er beeinflusst die Overflow-Flag im Prozessor-Statusregister:

```
POLL: BIT RDFLAG
      BVC POLLF ;Warten,
              bis Bit 6 = 1
```

Ein wesentlicher Nachteil des Polling-Verfahrens besteht darin, daß es die CPU blockiert: Sie kann nur dann innerhalb kürzester Frist auf eine Statusänderung reagieren, wenn sie innerhalb der Abfrageschleife keine anderen Programmschritte ausführt. Es gibt aber auch für dieses Problem eine Lösung: Der Flankendetektor läßt sich so programmieren, daß bei einer Pegeländerung an der Leitung PA7 ein Interrupt-Request (IRQ) erfolgt. Die Interrupt-Anforderung wird gleichzeitig mit dem Setzen der PA7-Flag abgegeben, indem der RIOT die IRQ-Leitung der CPU auf log. 0 zieht.

Der Flankendetektor wird programmiert, indem die CPU beliebige Daten in eines der Steuerregister schreibt. Vier solcher Register sind vorhanden:

- EDETA: negative Flanke detektieren
- EDETB: positive Flanke detektieren
- EDETC: negative Flanke detektieren und IRQ auslösen

EDETD: positive Flanke detektieren und IRQ auslösen

Countdown

Vor allem bei Steuerungsaufgaben spielen Zeitabläufe eine wichtige Rolle. Der Einsatz von Verzögerungsschleifen, in denen sich der Prozessor mit einer Anzahl sinnloser Instruktionen aufhält, ist aus mancherlei Gründen nachteilig, vor allem dann, wenn die Rechenzeit dringend benötigt wird. Der Countdown-Timer des RIOT 6532 ermöglicht es, den Computer sinnvoller einzusetzen als zum Abzählen von Zeiteinheiten. Er läuft im Systemtakt unabhängig von der CPU und meldet nach Ablauf einer vorher programmierten Zeitspanne den 'Time Out'.

Dieses Ergebnis wird ebenfalls durch das Setzen einer Statusflag angezeigt. Bit 7 des Registers RDFLAG ist das Timer-Statusbit. Ähnlich wie bei Anwendung des Flankendetektors kann der RIOT-Baustein aber auch so programmiert werden, daß er gleichzeitig mit dem Setzen der Timer-Flag einen IRQ auslöst.

Der Timer ist ein 8-Bit-Zähler, so daß maximal 255 Zeiteinheiten vorprogrammiert werden können. Zusätzlich hat der Programmierer die Wahl zwischen vier verschiedenen langen Zeiteinheiten, nämlich 1xT, 8xT, 64xT und 1024xT, wobei T für die Dauer eines Taktzyklus steht. In einem Standard-6502-System beträgt die Taktfrequenz gewöhnlich 1 MHz, T ist also 1 µs. Der Multiplikationsfaktor ergibt sich aus der Wahl des Registers, in das die CPU die gewünschte Anzahl von Zeiteinheiten bis zum Time Out schreibt. Die Wahl des Registers bestimmt auch, ob der RIOT nach dem Countdown einen Interrupt auslöst. Dies sind die Timer-Register und ihre Funktionen:

- CNTA: 1xT
- CNTB: 8xT
- CNTC: 64xT
- CNTD: 1024xT
- CNTE: 1xT, IRQ
- CNTF: 8xT, IRQ
- CNTG: 64xT, IRQ
- CNTH: 1024xT, IRQ

Ein Timer-Start könnte beispielsweise folgendermaßen ablaufen:

```
LDA #10
STA CNTG
```

Mit dem Schreiben des sogenannten 'Timer-Offsets' in das Register CNTG wird die Timer-Flag im Register RDFLAG zu-

128 Byte RAM	xx00...xx7F
PAD	xx80
PADD	xx81
PBD	xx82
PBDD	xx83
RDTDIS	xxD4
RDFLAG	xxD5
RD TEN	xxDC
EDETA	xxE4
EDETB	xxE5
EDETC	xxE6
EDETD	xxE7
CNTA	xxF4
CNTB	xxF5
CNTC	xxF6
CNTD	xxF7
CNTE	xxFC
CNTF	xxFD
CNTG	xxFE
CNTH	xxFF

Tabelle 1. Typische Adressenlage der RIOT-Register

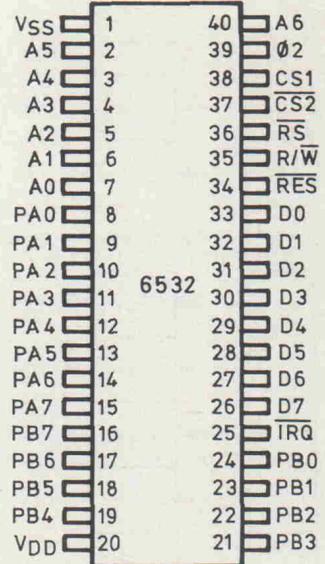


Bild 3. Pinbelegung des RIOT

rückgesetzt. Alle 64 µs nach diesem Vorgang zählt der RIOT eine Zeiteinheit ab. Beim Zählerstand 00 ist der Time Out erreicht. Einen Taktzyklus später ist die Timer-Flag gesetzt, und die IRQ-Leitung wird auf log. 0 gezogen. Die CPU erhält damit einen Interrupt Request, der allerdings nur dann zur Ausführung einer Interrupt-Routine führt, wenn die I-Flag im Prozessor-Statusregister zurückgesetzt ist.

Während der Timer-Laufzeit kann die CPU die verbleibende Anzahl von Zeiteinheiten in einem der Register RD TEN oder RDTDIS lesen. Liest die CPU das Register RDTDIS, so wird damit ein IRQ untersagt. Liest sie das Register RD TEN, so wird damit der IRQ erlaubt — unabhängig von der vorherigen Betriebsart.

Um die Timer-Flag in Polling Mode zu testen, eignet sich wiederum eine BIT-Instruktion:

```
TFTEST: BIT RDFLAG ;Timer-Flag gesetzt?
        BPL TFTEST ;Nein: Warten
```

Das folgende Programm dient dazu, die Länge eines Impulses zu messen:

```
START: SEC
      LDA #FF
WAIT: BIT PAD
      BMI START ;Warte auf
                L-Impuls an PA7
      STA CNTx ;Starte Timer
WAITB: BIT PAD
      BPL PAD ;Warte auf H-Pegel
      SBC RDTDIS
      STA LENGTH ;Länge ablegen
```

Ein wichtiger Hinweis für die Anwendung der Interrupt-Technik: Während der Ausführung einer Interrupt-Routine muß die Ursache des Interrupt

beseitigt werden, sonst bleibt die IRQ-Leitung auf L-Pegel, und bei der Rückkehr wird zugleich die nächste Unterbrechung ausgelöst. Der RIOT gibt die IRQ-Leitung frei, wenn die CPU den Timer neu startet (wahlweise für IRQ beim nächsten Time Out) oder RDTDIS liest. Wurde der Interrupt durch den Flankendetektor ausgelöst, so wird durch Lesen von RDFLAG die IRQ-Leitung freigegeben; gleichzeitig wird die PA7-Flag zurückgesetzt. Es empfiehlt sich, den Flankendetektor auch beim erstmaligen Einsatz in einem Programm durch Lesen von RDFLAG zu initialisieren.

Adressenlage

Welche effektiven Speicheradressen die verschiedenen RIOT-Register haben, hängt natürlich vom Anschluß des Bausteins ab. Der RIOT besitzt sieben Adreßleitungen, die in der Regel mit den niederwertigen Prozessor-Adreßleitungen verbunden werden. Verwendet man die Adreßleitung A7 — wie beim CEPAC 65 — zur Ansteuerung des Pins 36 (RAM-Select), dann ergibt sich die in Tabelle 1 aufgeführte Adressenlage. Dabei belegt der RIOT insgesamt eine Speicherseite (256 Byte); die untere Hälfte davon enthält das RAM, die obere die Steuer- und Portregister. Auf welcher Speicherseite der Baustein liegt, hängt wiederum von der übrigen Adreßdekodierung des Systems ab, aus der das Chip-Select-Signal gebildet wird. □