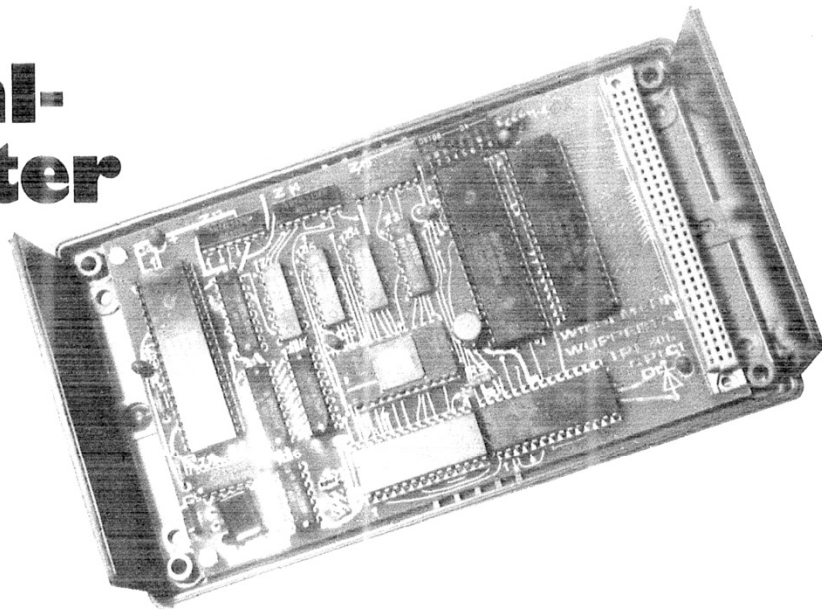


Reinhard Wiesemann

Minimal-computer mit Z80-CPU



Nachdem wir in Heft 2/81 einen Minimalcomputer mit der CPU 6504 vorgestellt haben, folgt jetzt ein ähnliches System mit dem Z80. Gegenüber seinem Vorgänger bietet es mehr Ein-/Ausgabe-Leitungen und einen größeren RAM-Bereich. Auch die Programmspeicherkapazität ist höher. Geeignete Systeme für die Programmentwicklung sind Tischcomputer mit der CPU Z80 – z. B. Nascom, TRS-80 und Video-Genie.

In kleinen Steuerungen, Schnittstellen und ähnlichen Anwendungen lassen sich Mikrocomputer aufgrund ihrer hohen Leistungsfähigkeit und Flexibilität mit Vorteil einsetzen. Für diese Bereiche werden bereits zahlreiche Systeme angeboten, die im Baukastenprinzip an das spezielle Problem angepaßt werden können: Aus einem meist umfangreichen Angebot an CPU-, Speicher- und Ein-/Ausgabe-Platinen stellt man sich das gewünschte System zusammen. Der Vorteil dieses Prinzips liegt in der leichten und praktisch unbegrenzten Erweiterbarkeit eines so aufgebauten Mikrocomputers.

Nachteilig ist, daß kleine Systeme nahezu nicht realisierbar sind: Ein minimales System benötigt meist bereits drei Computerplatinen (CPU, Speicher, Ein-/Ausgabe), eine Buskarte sowie ein Netzteil mit meist mehreren Versorgungsspannungen. Berücksichtigt man dazu das dann häufig notwendige 19-Zoll-Gehäuse, so kann das so aufgebaute System in vielen Anwendungen schon allein aus preislichen Gründen nicht eingesetzt werden.

Der hier beschriebene Minimal-Mikrocomputer vermeidet diese Nachteile, da er speziell für den Einsatz in kleinen Systemen entwickelt wurde: Der gesamte Computerteil (inklusive CPU, RAM, EPROM, Ein-/Ausgabe) befindet sich auf einer einzigen Platine, und es ist nur eine 5-V-Versorgungsspannung nötig. Auf diese Weise können kleine Mikrocomputer sehr kompakt und preiswert in Standardgehäuse eingebaut werden. Ein vollständiges System benötigt neben der Computerplatine nur eine zusätzliche Karte, die die anwendungsspezifische Schaltung (Relais, Treiber usw.) und das Netzteil beinhaltet. Bild 1 zeigt die Schaltung des Einplatinencomputers MMC-1. In den Bildern 2 und 3 sind Platinenlayout und Bestückungsplan dargestellt. Die Platine ist unbestückt oder fertig bestückt vom Autor beziehbar (Postfach 20 16 05, 5600 Wuppertal 2).

Schaltungsdetails

Beim Einschalten der Versorgungsspannung wird die CPU durch das aus R8 und C7 bestehende RC-Glied zurückge-

setzt. Der $\overline{\text{INT}}$ -Eingang kann von einer speziellen Anwenderschaltung frei benutzt oder über C10 mit dem Baud-Rate-Generator verbunden werden. In diesem Fall erhält die CPU alle 256 Taktzyklen ein INT-Signal. Da auf der Platine keine Interrupt-Vektoren erzeugt werden, muß Interrupt-Mode 1 (Restart bei 038H) programmiert werden.

Die Karte bietet die Möglichkeit, bis zu 2 KByte EPROM (1 × Typ 2716 – Z10) sowie 2 KByte RAM (4 × Typ 2114 – Z2...Z5) zu bestücken. Die acht niedrigwertigen Bits des Adreßbusses sowie der Datenbus sind durch die Bausteine Z16...Z18 gepuffert.

Z13 arbeitet als Speicher-Adreßdecoder und kann bis zu acht Speicherblöcke zu jeweils 2 KByte adressieren. Das EPROM sowie jeder RAM-Block belegen jeweils einen Bereich, so daß sich folgende Speicheraufteilung ergibt:

2400H	FFFFH	frei
2000H	23FFH	RAM 2 Z2, Z3 1 KByte
1400H	1FFFH	frei
1000H	13FFH	RAM 1 Z4, Z5 1 KByte
800H	FFFFH	frei
0H	7FFFH	EPROM Z10 2 KByte

Obwohl die Blockadressierung in Schritten von 2 KByte vorgenommen wird, liegt zwischen EPROM und dem ersten RAM ein Bereich von 2 KByte. Auf diese Weise ist es sehr einfach möglich, statt des EPROMs 2716 den Typ 2732 (mit doppelter Kapazität) zu verwenden (+5-V-Leitung zu Anschluß 21 auf der Platine auftrennen und Anschluß 21 an Adresse 11 legen).

Die Platine arbeitet mit einer Taktfrequenz von 2,4576 MHz, die durch den Baustein Z11 erzeugt wird. Diese Frequenz erlaubt einen besonders einfachen Aufbau des Baud-Rate-Generators: Ein Binärteiler (Z12) stellt die benötigten Frequenzen an Steckbrücken zur Verfügung:

Anschluß	Steckbrücke	Frequenz
13	1	153,6 kHz
11	2	76,8 kHz
10	3	38,4 kHz
9	4	19,2 kHz
8	5	9,6 kHz

Durch Wahl einer der fünf Brücken wird die entsprechende Frequenz an die Send- und Empfangsregister der UARTs (Z8 und Z9) gelegt. Die Baud-Rate jedes Kanals ergibt sich aus dieser Frequenz sowie einem Teilerfaktor, der per Software programmiert werden kann. Dieser Faktor kann 1, 16 oder 64 sein, so daß die Baud-Rate im Bereich von 150 Bd (= 9,6/64) bis 153 600 Bd (= 153 600/1) liegen könnte. Erlaubt sind jedoch nur Geschwindigkeiten bis 9600 Bd (Tabelle 1). In den meisten Fällen bietet es sich an, den Faktor 16 zu programmieren, da damit die üblichen Geschwindigkeiten einstellbar sind. Durch Wahl unterschiedlicher Faktoren oder durch Einbau eines zweiten Satzes von Steckbrücken sind unterschiedliche Baud-Raten für beide UARTs möglich.

Ein-/Ausgabe-Möglichkeiten

Z6 und Z7 arbeiten als PIO-Bausteine und stellen dem Anwender insgesamt bis zu 48 TTL-kompatible Leitungen zur Verfügung, die als Ein- oder Ausgang zur Steuerung von Relais usw. bidirektional oder mit Handshake-Signalen benutzt werden können. Beim Einschalten der Versorgungsspannung werden diese Bausteine durch das aus R9 und C8 bestehende RC-Glied zurückgesetzt.

Der Baustein Z15 erlaubt die Abfrage von sechs DIL-Schaltern per Programm. Ein geschlossener Schalter legt das an diesem Eingabekanal liegende Bit auf „0“, bei geöffnetem Schalter liegt über Pull-up-Widerstände „1“ an.

Z14 arbeitet als Adreßdecoder für sämtliche Ein-/Ausgabe-Bausteine. Da die niedrigwertigen zwei Adreßbits von den PIO-Bausteinen für interne Adressierungen verwendet werden, decodiert dieser Baustein die Adressen 2...4. Von den damit möglichen acht I/O-Adressen sind sechs belegt. Sie ergeben zusammen mit der internen Decodierung der PIOs und UARTs den in Tabelle 2 dargestellten Ein-/Ausgabe-Bereich.

Alle Ein-/Ausgabe-Signale sowie die Versorgungsspannung sind an eine 64polige VG-Leiste geführt. An dieser Leiste können die anwenderspezifische Schaltung sowie das Netzteil aufgesteckt werden (Tabelle 3). Die Bezeichnungs-

Tabelle 1: Die Baud-Rate der UARTs ergibt sich aus Taktfrequenz und programmiertem Faktor

Brücke	Faktor	Baud-Rate
1	1	nicht benutzen
1	16	9600 Baud
1	64	2400 Baud
2	1	nicht benutzen
2	16	4800 Baud
2	64	1200 Baud
3	1	nicht benutzen
3	16	2400 Baud
3	64	600 Baud
4	1	nicht benutzen
4	16	1200 Baud
4	64	300 Baud
5	1	9600 Baud
5	16	600 Baud
5	64	150 Baud

weise der einzelnen Signale entspricht weitestgehend der in den Datenbüchern üblichen Art. Da die Bausteine 8255 und 8251 jeweils zweimal vorhanden sind, gibt eine nachgesetzte Ziffer an, um welchen es sich handelt (/1 = Z6 bei PIO-Signalen; Z8 bei UART-Signalen; /2 = Z7 bei PIO-Signalen; Z9 bei UART-Signalen).

Die Funktion der Stifte a4 und c4 ist nicht festgelegt. Je nach Anwendung

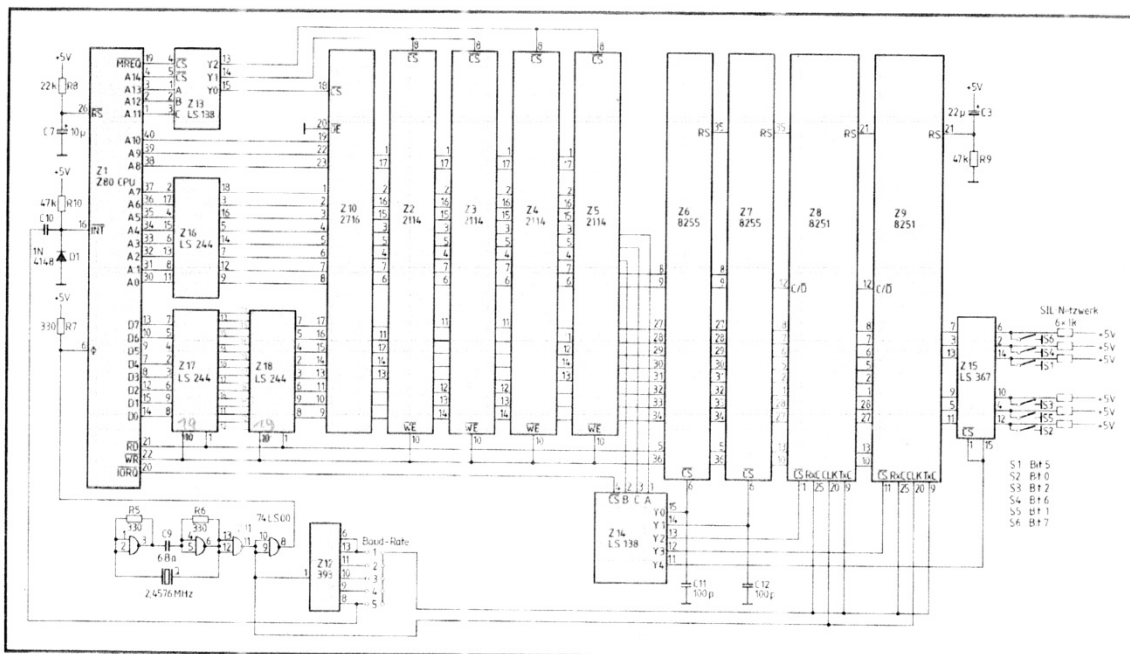


Bild 1. Schaltung des Z80-Einplatinencomputers

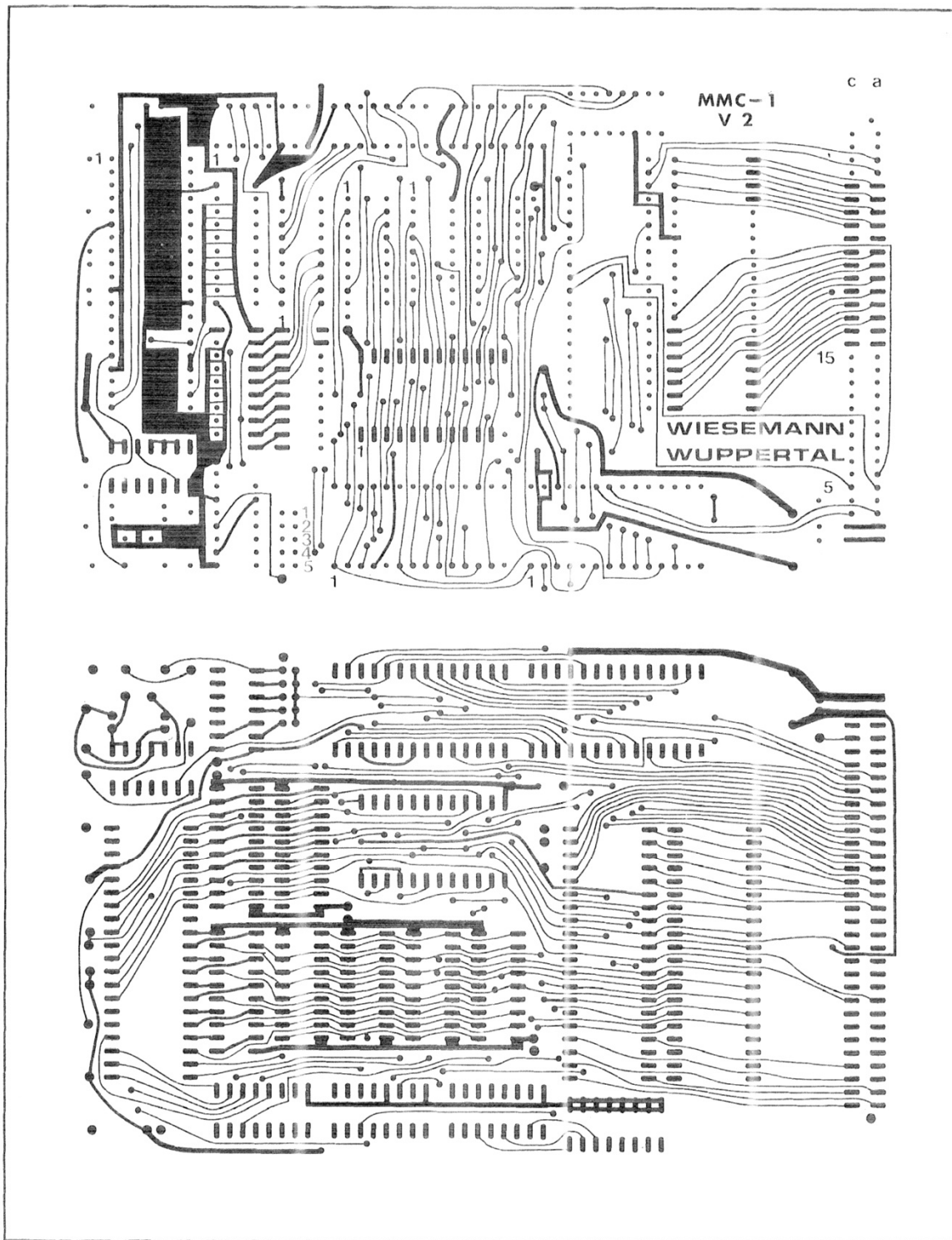


Bild 2. Platinenlayout

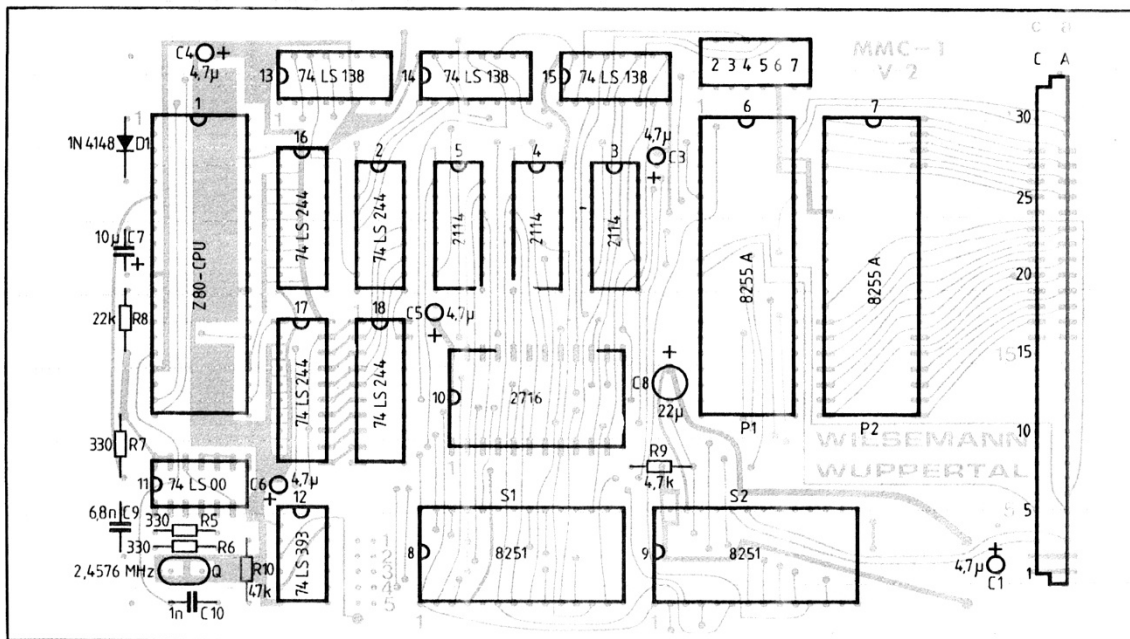


Bild 3. Bestückungsplan. Das SIL-Netzwerk wird auf der Lötseite bestückt (Switch-In-Line)

können hier z. B. die Signale $\overline{DSR1}$ und $\overline{RTS1}$ oder die $\overline{INT}$ und Takt-Leitung angeschlossen werden, indem eine Brücke von den entsprechenden Punkten an diesen Anschluß gelegt wird:

$\overline{DSR1}$ = Stift 22 von Z8. Dieser Anschluß ist auf der Lötseite der Platine mit Stift 26 (+5 V) verbunden. Soll dieser Stift als Eingang benutzt werden, so ist diese Verbindung aufzutrennen.

$\overline{RTS1}$ = Stift 23 von Z8. Dieser Anschluß liegt auf der Platine frei, so daß er direkt benutzt werden kann.

$\overline{INT}$ = Stift 16 der CPU – liegt über R10 als Pull-up-Widerstand an +5 V. Dieser Eingang kann direkt benutzt werden. Als Schutz vor negativen Eingangsspannungen dient D1.

Takt = Stift 1 von Z12 (2,4576 MHz).

Hinweise zur Programmierung

Zur Programmierung des MMC-1 sind die Datenblätter der Bausteine 8251 und 8255 (z. B. im Intel-Datenbuch) sowie das Z80-Manual (Zilog) notwendig. Nach dem Anlegen der Versorgungsspannung werden μP und I/O-Bausteine durch getrennte RC-Glieder zurückgesetzt. Der μP legt Adresse 0 an den

Adreßbus und spricht damit die erste EPROM Adresse an. Die Parallelports der I/O-Bausteine 8255 werden in Mode 0 (Input) geschaltet. Um zu vermeiden, daß der μP (wenn seine Reset-Zeit kürzer als die der I/O-Bausteine ist) zu früh die I/O-Bausteine anspricht, sollte am Anfang des Programms eine Verzögerungsschleife vorgesehen werden. Zum Beispiel:

0000		LD	A, 08H
0002	V1	LD	C, 0FFH
0004	V2	LD	B, 0FFH
0006	V3	DJNZ	V3
0008		DEC	C
0009		JR	NZ, V2
000B		DEC	A
000C		JR	NZ, V1

Bei Verwendung von Interrupts sollte zuvor der Stackpointer initialisiert sein. Die Lage des Stacks im RAM ist beliebig; beachten Sie, daß er in Richtung auf niedrigere Adressen wächst. Die Platine erlaubt nur den Betrieb im Interrupt-Mode 1 des Z80. In dieser Betriebsart führt die CPU nach Erhalt eines Interrupt einen RST-Befehl nach Adresse 038H aus. An dieser Stelle kann das Interrupt-Unterprogramm programmiert werden. Zur Rückkehr aus dem Interrupt-Programm reicht ein RET-Befehl aus (RETI ist nicht notwendig). Die Verwendung der $\overline{INT}$ -Leitung des Z80 ist beliebig (ein Pull-up-Widerstand befindet sich auf der Platine, so daß eine externe Schaltung z. B. mit Open-Collec-

Tabelle 2: Ein-/Ausgabe-Bereich der PIOs und UARTs

15H	8251/2	Control	(8251/2 = Z9 auf MMC-Platine)
14H	8251/2	Data	
13H	8255/2	Control	(8255/2 = Z7 auf MMC-Platine)
12H	8255/2	Port C	
11H	8255/2	Port B	
10H	8255/2	Port A	
8H	74LS367	(Z15)	(DIL-Schalter 1...6) – diese Adresse nur lesen
5H	8251/1	Control	(8251/1 = Z8 auf MMC-Platine)
4H	8251/1	Data	
3H	8255/1	Control	(8255/1 = Z6 auf MMC-Platine)
2H	8255/1	Port C	
1H	8255/1	Port B	
0H	8255/1	Port A	

tor-Ausgang angeschlossen werden kann) – diese Leitung kann über eine kurze Brücke an einen der freien Stifte auf der VG64-Leiste gelegt werden. Wird im Programm eine Zeitbasis benötigt, so kann durch Einbau von C10 ein Interrupt-Signal alle 256 Taktzyklen erzeugt werden (alle 0,1042 ms).

Programmbeispiele

Einige Unterprogramme können in vielen Anwendungen des MMC-1 unverändert übernommen werden. Alle nachfolgenden Unterprogramme (Bilder 4 und 5) kehren mit gesetztem Z-Flag zurück, wenn die entsprechende Ein- oder Ausgabe nicht möglich war (Eingabe: Es ist kein Zeichen empfangen worden; Ausgabe: Das zuvor gesendete Zeichen ist von der Peripherieschaltung noch nicht abgerufen worden). Auszugebende Zeichen müssen in Register C liegen, empfangene Zeichen gelangen in Register C.

Tabelle 3: Signale an der 64poligen VG-Leiste des MMC-1

	Reihe a	Reihe c
1	GND	GND
2	+5 V	+5 V
3	DSR2	RTS2
4	sh. Text	sh. Text
5	PC7/1	PC6/1
6	PC4/2	DTR2
7	DTR1	CTS1
8	CTS2	T × D 2
9	T × D 1	R × D 2
10	R × D 1	PB2/1
11	PB1/1	PB0/1
12	PC3/1	PC2/1
13	PC0/1	PC4/1
14	PC5/1	PC1/1
15	PB3/1	PB4/1
16	PB5/1	PB6/1
17	PB7/1	PB3/2
18	PB4/2	PB5/2
19	PB6/2	PB7/2
20	PB2/2	PC5/2
21	PB1/2	PB0/2
22	PC3/2	PC2/2
23	PC1/2	PC0/2
24	PC6/2	PC7/2
25	PA7/2	PA0/2
26	PA6/2	PA1/2
27	PA5/2	PA2/2
28	PA4/2	PA3/2
29	PA6/1	PA7/1
30	PA5/1	PA0/1
31	PA1/1	PA4/1
32	PA2/1	PA3/1

Die Programme verändern den Inhalt der Register A und F. Vor Aufruf muß der entsprechende Baustein für diese Betriebsart vorbereitet – d. h. ein PIO-Bau-

stein muß zuvor den entsprechenden Betriebsartbefehl, ein UART-Baustein den Betriebsartbefehl und ein Kommando erhalten haben.

```

;Eingabe mit Strobe über Port A in Z6
PA1IN      IN        A, (PC1)
           BIT        5, A           ;Bit 5 = 1, wenn Daten
                                       ;empfangen wurden
           RET        Z           ;Z-Flag, wenn keine Daten
           LD         C, A           ;Sonst Rückkehr mit Daten
           RET                                     ;in Register C

;Ausgabe mit Strobe über Port B in Z6
PB1OUT      IN        A, (PC1)
           BIT        1, A           ;Bit 1 = 1, wenn Daten
                                       ;gesendet werden dürfen
           RET        Z           ;Wenn nicht, zurück mit
                                       ;Z-Flag
           LD         A, C           ;Sonst Daten aus Register
           OUT        (PB1), A       ;C ausgeben
           RET

```

Bild 4. Unterprogramme für die parallele Ein-/Ausgabe

```

;Serielle Eingabe über Z8
S1IN        IN        A, (SC1)
           BIT        1, A           ;Bit 1 = 1, wenn Daten
                                       ;empfangen wurden
           RET        Z           ;Wenn nicht, zurück mit
                                       ;Z-Flag
           IN         A, (SD1)       ;Sonst Daten einlesen
           RET

;Serielle Ausgabe über Z8
S1OUT       IN        A, (SC1)       ;Bit 0 = 1, wenn Daten
           BIT        0, A           ;ausgegeben werden
                                       ;können
           RET        Z           ;Wenn nicht, zurück mit
                                       ;Z-Flag
           LD         A, C           ;Sonst Daten aus Register
           OUT        (SC1), A       ;C ausgeben
           RET

```

Bild 5. Unterprogramme für die serielle Ein-/Ausgabe

Ist Ihre EMUF-Anwendung „top secret“?

Der in mc 1981, Heft 2, beschriebene „Einplatinen-Mikrocomputer für universelle Festprogramm-Anwendung“ (EMUF) eignet sich für tausend Dinge – oder sogar noch mehr. Vielleicht realisieren Sie mit der EMUF-Platine gerade eine allgemein interessierende Anwendung? Rufen Sie uns an oder schreiben Sie uns! Wir freuen uns über jede neue EMUF-Idee, und vielleicht sind andere Leser ganz gierig darauf, Ihre Anwendung kennenzulernen. Oder ist Ihre EMUF-Applikation etwa „top secret“?

Ihre mc-Redaktion