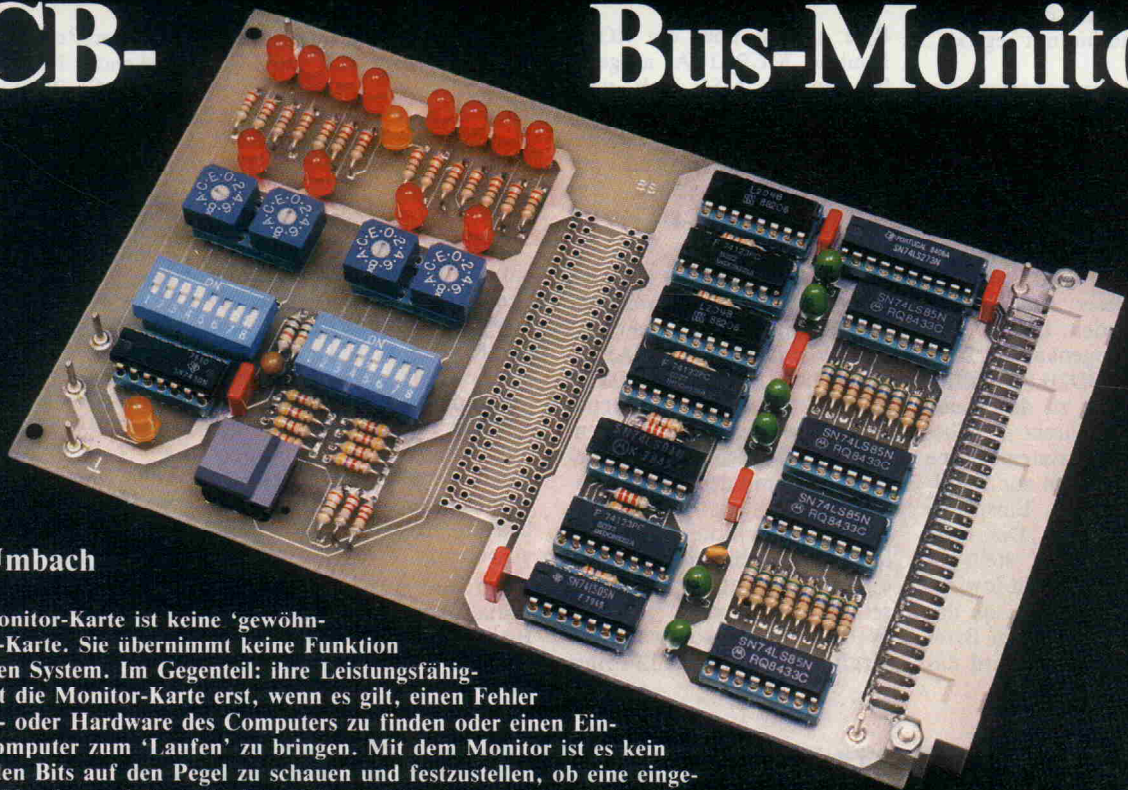


Dem Fehler auf der Spur:

ECB-Bus-Monitor

Georg Umbach

Die Bus-Monitor-Karte ist keine 'gewöhnliche' ECB-Karte. Sie übernimmt keine Funktion im laufenden System. Im Gegenteil: ihre Leistungsfähigkeit beweist die Monitor-Karte erst, wenn es gilt, einen Fehler in der Soft- oder Hardware des Computers zu finden oder einen Einplatinen-Computer zum 'Laufen' zu bringen. Mit dem Monitor ist es kein Problem, den Bits auf den Pegel zu schauen und festzustellen, ob eine eingestellte Kombination auf den Adreß- und Steuerleitungen erreicht wird.



Als 'Testkarte' ist der ECB-Monitor in erster Linie für Computer-Anwender interessant, die sich mit Computer-Hardware sowie der Programmierung beschäftigen. Aber auch beim Erstellen von Programmen für Z80-Einplatinen-Computer, wie zum Beispiel dem EPAC-80 aus c't 7/84, leistet die Monitor-Karte gute Dienste. Allerdings: ohne Grundlagenwissen kann man die Monitor-Karte nicht sinnvoll einsetzen.

Schaltungsdetails

Die sechs wichtigsten Steuerleitungen vom ECB-Bus (MI, MREQ, IORQ, RD, WR, RFSH) sind auf die Inverter 74LS05 (IC6) geführt. Mit dem am Ausgang von IC6 angeschlossenen DIL-Schalter (S1) werden die Signale 'Wired-AND' verknüpft. Das so gewonnene Steuersignal gelangt auf die '='-Eingänge der vier

Adreß-Komparatoren (IC2 bis IC5).

Jeder Komparator vergleicht vier Adreßleitungen mit einem durch die HEX-Schalter (HEX 1 bis HEX 4) vorgegebenen Wert. Stimmen die mit S1 gewählten Steuersignale mit denen am Bus überein und liegen zusätzlich noch die mit dem HEX-Schalter gewählten vier Adreß-Nibbles auf dem Bus, so wird das dem Komparator nachgeschaltete Monoflop getriggert, und die LED für diesen Adreßteil leuchtet etwa eine halbe Sekunde lang auf. Diese Verlängerung ist deshalb nötig, weil das gewonnene Steuersignal nur wenige Mikrosekunden lang ansteht und somit kaum sichtbar wäre.

Die '='-Ausgänge der vier Adreßkomparatoren sind zusätzlich noch über einen DIL-Schalter (S2) auf die Eingänge des NAND-Gatters 74LS30 (IC7) geschaltet. Mit dem DIL-Schalter kann man bestimmen,

welche Adreßteile gleichzeitig vorhanden sein müssen, damit der Ausgang des Gatters IC7 auf logisch 0 geht. Dieser Ausgang dient als Trigger-Quelle für ein Oszilloskop.

Solange der mit den Schaltern vorgewählte logische Zustand der Bus-Signale ansteht, liegt das Trigger-Signal auf logisch 0. Verändern sich die Bus-Signale (die Steuersignale zuerst), wird das momentan anliegende Bitmuster der acht Datenleitungen in das Register 74LS273 (IC1) mit der positiven Flanke des Trigger-Signals eingeschrieben. Der Zustand der Datenleitungen bleibt so erhalten und wird über acht Leuchtdioden (LD1 bis LD8) bis zum Erreichen des nächsten vorgewählten Buszustands gespeichert und angezeigt.

Für die Triggerung eines Oszilloskops und die Speicherung des Datenbus-Wortes kann man auch nur die Steuersignale verwenden. Dies ist zum Bei-

spiel dann sinnvoll, wenn man die Interrupt-Vektoren von Peripheriebausteinen darstellen will.

Mit einem weiteren DIL-Schalter (S1) kann man eine WAIT-Logik aktivieren. Die negative Flanke eines Trigger-Impulses setzt ein RS-Flipflop, das aus zwei NAND-Gattern (IC11) besteht. Dieses Flipflop erzeugt ein WAIT-Signal für den (ECB-)Bus. Mit einem 'Digi-Taster' (T1), der durch zwei NAND-Gatter von IC11 entprellt ist, wird ein Monoflop getriggert, das das RS-Flipflop zurücksetzt und somit den WAIT-Zustand beendet. Das ECB-Bus-Signal 'Powerclear' (Leitung 26c) setzt die WAIT-Logik beim Einschalten des Computers zurück.

Während dieser WAIT-Zeit sind alle Signalzustände 'eingefroren' und können sogar mit einem Multimeter überprüft werden. Allerdings gibt die CPU in diesem Zustand keine

Refresh-Adressen aus, so daß dynamische Speicher, die den Refresh von der CPU benötigen, ihren Speicherinhalt recht schnell verlieren. Bei Computern mit statischen Speichern, wie zum Beispiel dem EPAC-80, kann man die WAIT-Logik gefahrlos anwenden. Aber auch bei Systemen, die den Speicher-Refresh ohne Hilfe der CPU durchführen, kann man die WAIT-Logik einsetzen.

Aufgebaut

Im Layout der Monitor-Karte ist eine Trennstelle vorgesehen, an der man die Karte bei Bedarf in zwei Abschnitte teilen kann. Die Kartenteile werden dann über ein 50poliges Kabel verbunden. Auf dem Teil der Karte, der direkt mit dem Bus verbunden wird, sind alle wesentlichen Bauteile untergebracht. Der andere Teil trägt die Bedienungs- und Anzeigelemente. Hier befindet sich auch der Abgreifpunkt für die Triggering eines Oszilloskops. Diese 'getrennte' Lösung hat den Vorteil, daß alle Bedienteile bequem zu erreichen sind, auch wenn man keinen Bus-Extender verwendet.

Man beginnt die Bestückung der Karte am besten mit den Widerständen; es folgen die Fassungen für die ICs und die DIL- beziehungsweise HEX-Schalter. In welche Lochreihe der Platine die Fassungen für die HEX-Schalter kommen, hängt von der Ausführung der Schalter ab: Bekommt man den Typ, der in der Stückliste angegeben ist, kommen die Fassungen in die beiden oberen Lochreihen. Häufig erhält man aber HEX-Schalter mit spiegelbildlicher Belegung, deren Fassungen dann in die unteren beiden Lochreihen eingesetzt werden müssen.

Anschließend bestückt man die Kondensatoren, bis auf C6 (33pF). Meistens ist der vom Monoflop IC9 erzeugte Impuls ausreichend lang, um die WAIT-Logik einwandfrei zurückzusetzen. Erst wenn hierbei Probleme auftreten, sollte man diesen Kondensator einsetzen. Ein zu langer Impuls verringert die Single-Step-Fähigkeit der Karte innerhalb eines Befehls bei Systemen mit Taktfrequenzen ab 2,5 MHz. Zuletzt sollte man die Leuchtdioden, Pfostenstecker, VG-

Leiste und den 'Digi-Taster' einlöten.

Nun kann man alle ICs bis auf IC11 einsetzen und die Funktionsfähigkeit der Karte prüfen. Dazu könnte man die 'Debugging'-Sitzungen wie im Abschnitt 'Beispielhaft' beschrieben nachvollziehen. Abschlie-

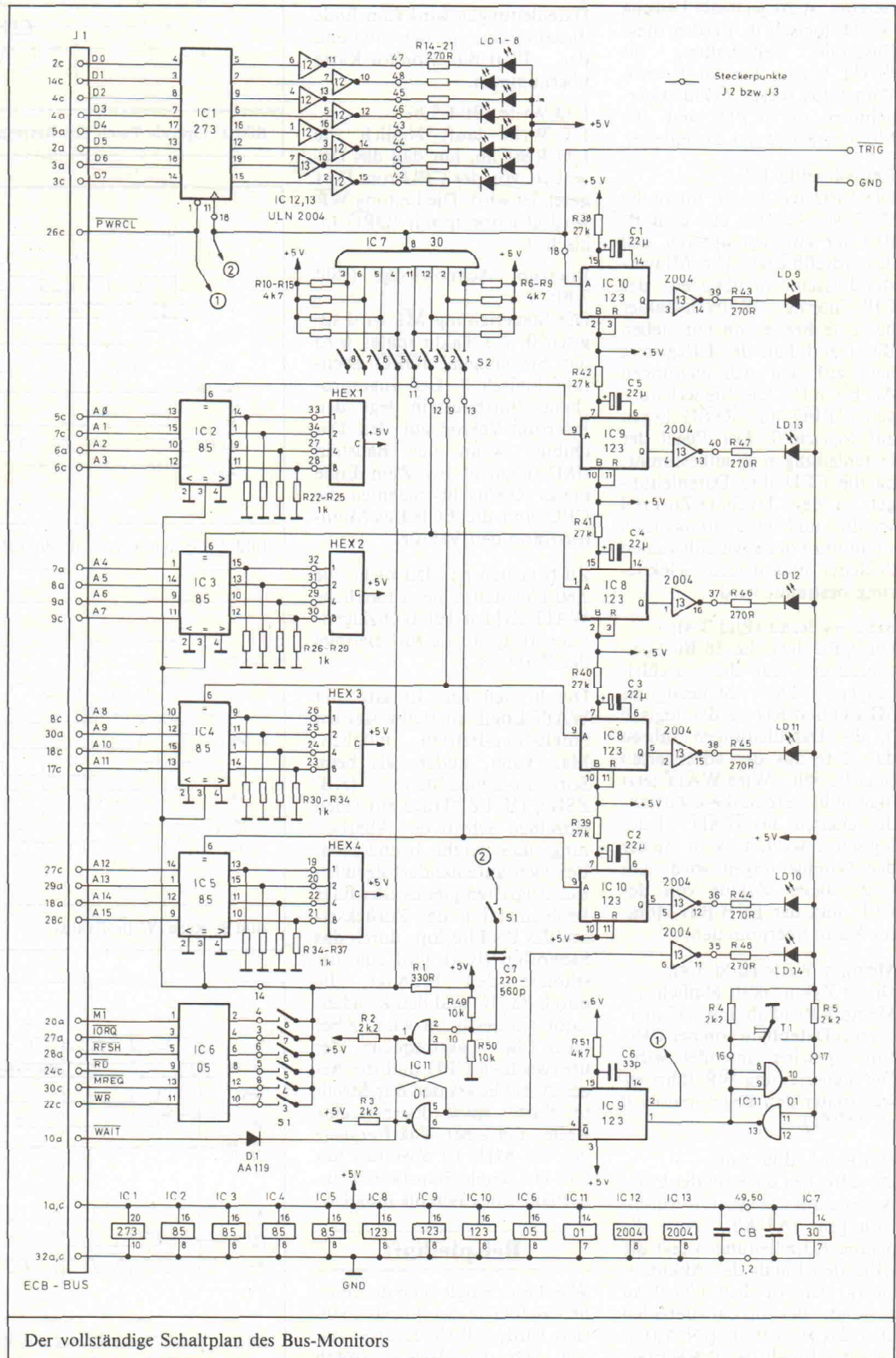
ßend wird auch IC11 (74LS01) eingesetzt und die WAIT-Logik an einem Computer mit statischem Speicher geprüft.

Sieben Zyklen

Wie schon eingangs erwähnt, kann die Monitor-Karte nur

dann sinnvoll eingesetzt werden, wenn der Anwender genaue Kenntnis über die Funktionsweise der Z80-CPU hat. Deshalb sollen die wichtigsten Signale des Prozessors kurz erklärt werden.

In den Bildern 1 bis 6 sind die Zusammenhänge zwischen den



CPU-Signalen und den Bus-Signalen dargestellt. Es gibt insgesamt sieben Bus-Zyklen:

Opcode Fetch (Bild 1,2):

Die CPU holt sich das erste Byte eines Befehls aus dem Speicher. Dabei sind die Leitungen $\overline{M1}$, $\overline{MREQ}$ und $\overline{RD}$ logisch 0. Die Datenleitungen führen das (erste) Byte eines Befehls. Wird jetzt die Leitung $\overline{WAIT}$ logisch 0, werden diese Zustände beibehalten, bis $\overline{WAIT}$ wieder logisch 1 wird. Zum Ende dieses Zyklus übernehmen die CPU und die Monitor-Karte das Datenbyte.

Refresh (Bild 1,2):

Die CPU legt eine 8 Bit breite (Refresh-)Adresse aus dem R-Register auf den unteren Teil des Adreßbusses. Um Mißverständnissen vorzubeugen: der CPU-interne Refresh-Zähler hat eine Breite von nur sieben Bit. Der Inhalt des I-Registers liegt auf den Adreßleitungen A8 bis A15. Die Steuerleitungen $\overline{MREQ}$ und $\overline{RFSH}$ liegen auf logisch 0. Der Pegel der Datenleitungen ist unbestimmt, da die CPU ihre Datenleitungen in den Tri-State-Zustand schaltet und der Leitungspegel somit nur vom angeschlossenen Speicher und dessen Selektierung bestimmt wird.

Memory Read (Bild 3,4):

Die CPU legt die 16-Bit-Speicheradresse auf die Adreßleitungen. Die Steuersignale $\overline{MREQ}$ und $\overline{RD}$ werden logisch 0; die Datenleitungen führen das Byte aus der adressierten Speicherzelle. Wird $\overline{WAIT}$ jetzt logisch 0, werden diese Zustände gehalten, bis $\overline{WAIT}$ wieder logisch 1 wird. Das Bitmuster der Datenleitungen wird zum Ende dieses Zyklus von der CPU und der ECB-Bus-Monitor-Karte übernommen.

Memory Write (Bild 3,4):

Dieser Zyklus läuft ähnlich wie Memory Read ab, nur daß hierbei das Datenbyte von der CPU zum Speicher gesendet wird. Die Steuerleitung $\overline{WR}$ führt etwas später den Pegel logisch 0 als $\overline{MREQ}$.

I/O-Read (Bild 5,6):

Die CPU legt die 8 Bit der Port-Adresse auf die unteren Adreßleitungen (A7-A0). Auf die oberen Adreßleitungen legt die CPU den Inhalt des Akkumulators, der vor dem I/O-Read bestand. Bei In/Out-Befehlen über das Register C (IN A,(C)) liegt der Inhalt des B-Registers

auf den oberen Adreßleitungen. Die Steuerleitungen $\overline{IORQ}$ und $\overline{RD}$ liegen auf logisch 0, auf den Datenleitungen liegt das aus dem Port gelesene Byte. Die CPU fügt automatisch einen $\overline{WAIT}$ -Zyklus ein. Liegt die $\overline{WAIT}$ -Leitung auf logisch 0, werden diese Zustände gehalten, bis $\overline{WAIT}$ wieder logisch 1 wird. Das Bitmuster der Datenleitungen wird zum Ende dieses Zyklus von der CPU und der ECB-Bus-Monitor-Karte übernommen.

I/O-Write (Bild 5,6):

I/O-Write läuft ähnlich wie I/O-Read ab, nur daß das Datenbyte von der CPU zum Port gesendet wird. Die Leitung $\overline{WR}$ wird gleichzeitig mit $\overline{IORQ}$ logisch 0.

Interrupt Acknowledge (Bild 7,8):

Die Steuerleitung $\overline{M1}$ wird logisch 0. 2,5 Takte später wird das Steuersignal $\overline{IORQ}$ ebenfalls logisch 0. Der angesprochene Portbaustein legt den Interrupt-Vektor auf den Datenbus, wenn der Baustein 'IM2-tauglich' ist. Zum Ende dieses Zyklus übernehmen die CPU und die ECB-Bus-Monitor-Karte den Vektor.

Zu beachten ist, daß nicht alle Z80-I/O-Bausteine zusätzliche $\overline{WAIT}$ -Zyklen bei I/O-Zugriffen vertragen, wie zum Beispiel der Z80-CTC.

Durch richtigen Einsatz der $\overline{WAIT}$ -Logik ist (teilweise) ein Single-Step-Betrieb möglich. Man kann, anders als beim Software-Single-Step (z.B. ZSID, DDTZ, Trace 80) jeden einzelnen Schritt der Abarbeitung eines Befehls beobachten. Bei zwei aufeinanderfolgenden Bus-Zugriffen gibt es allerdings Probleme: Für das Zurücksetzen des RS-Flipflops durch das Monoflop (IC9) wird eine bestimmte Zeit benötigt, die durch das IC und den Kondensator C6 bestimmt wird. Aber auch die Taktfrequenz der überwachten CPU und die Art des Anschlusses der Bus-Monitor-Karte spielt hierbei eine Rolle. Bei einer Taktfrequenz von 2,5 MHz ist aber fast immer ein Single-Step-Betrieb innerhalb eines Befehls möglich.

Beispielhaft

Wie dieser Single Step durch einen Befehl in der Praxis realisiert wird, soll ein Beispiel zeigen. Ab der Adresse 0844h

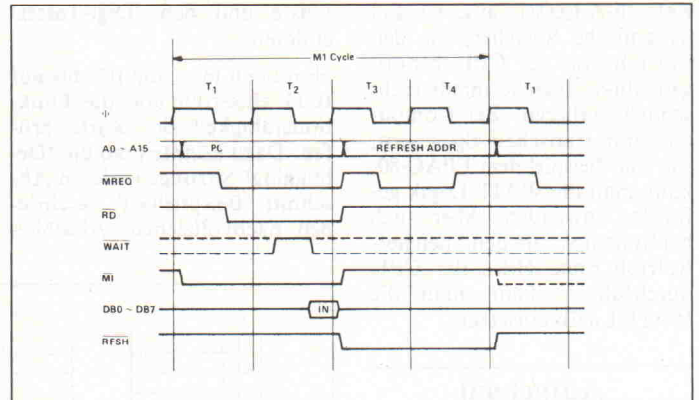


Bild 1. Opcode Fetch mit Refresh

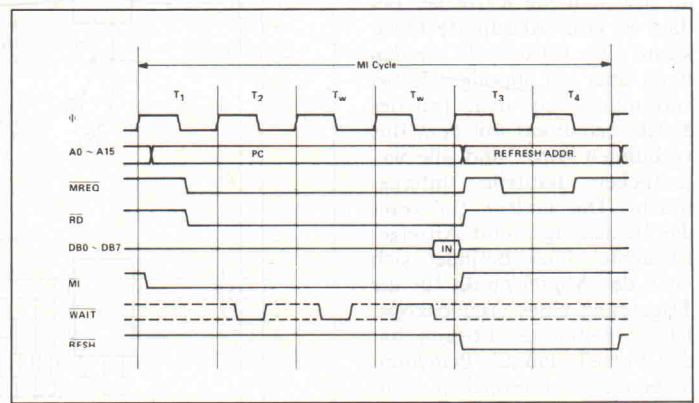


Bild 2. Opcode Fetch mit Wait-Zyklus und Refresh

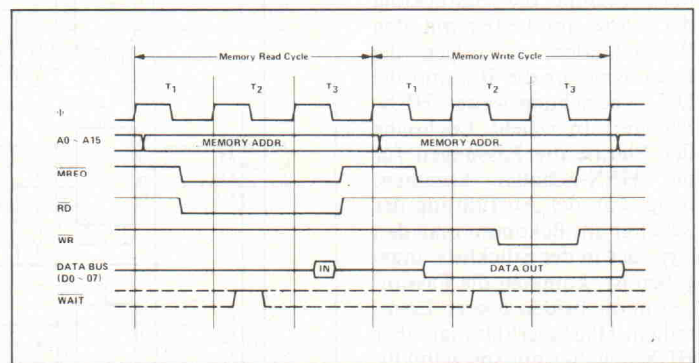


Bild 3. Read/Write-Zyklus

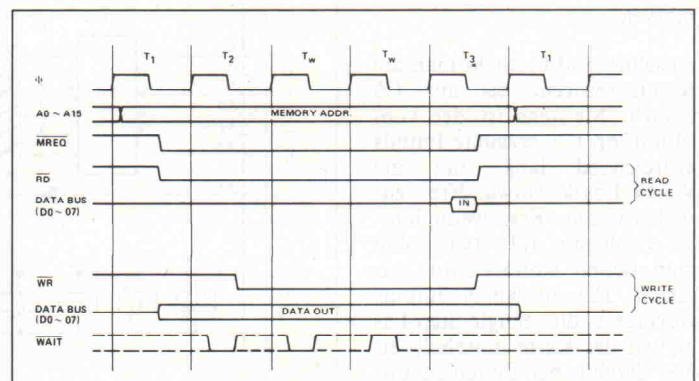


Bild 4. Read/Write-Zyklus mit Wait-Zyklus

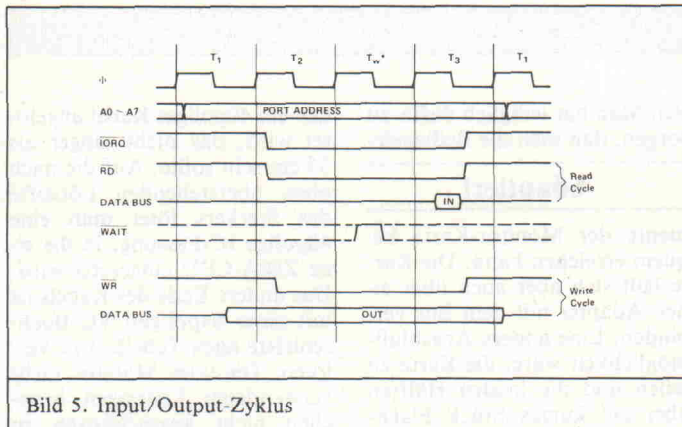


Bild 5. Input/Output-Zyklus

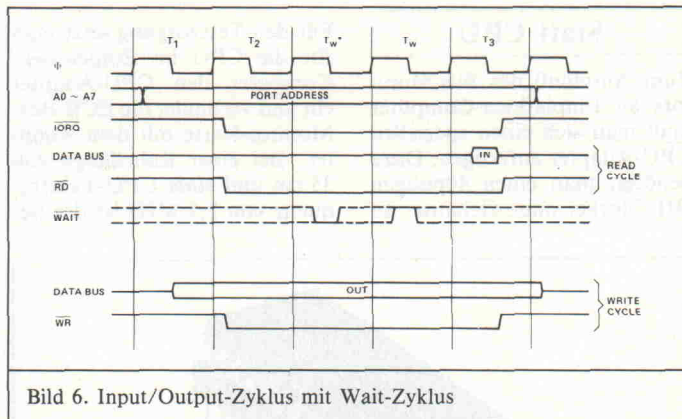


Bild 6. Input/Output-Zyklus mit Wait-Zyklus

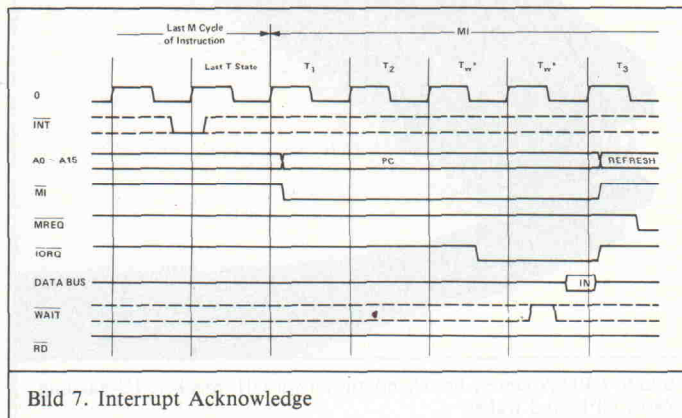


Bild 7. Interrupt Acknowledge

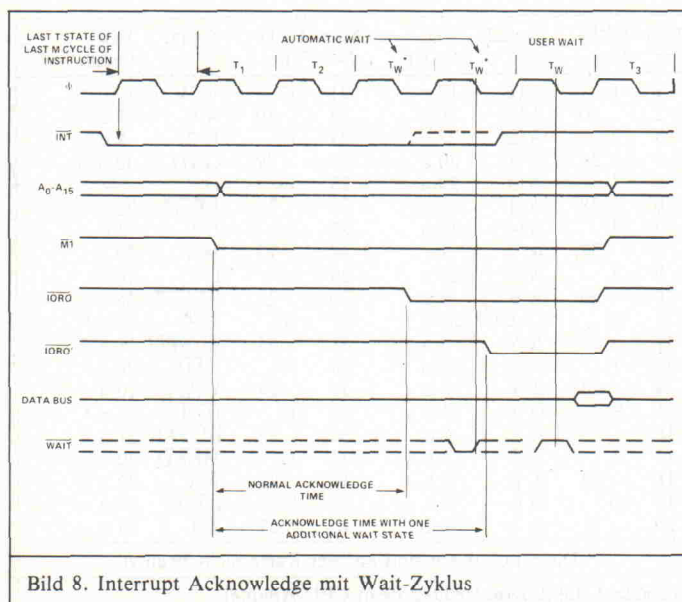


Bild 8. Interrupt Acknowledge mit Wait-Zyklus

steht im Speicher der folgende Befehl, wobei in den Speicherzellen 0EDCh / 0EDDh der Wert 3322 stehen soll:

0844 ED5BDC0E LD DE,(0EDCh)

Um durch diesen Befehl zu 'steppen', müssen die Schalter so gesetzt sein, daß der Computer beim ersten Opcode Fetch (ED) auf Adresse 0844 anhält. Hierzu sind die DIL-Schalter für MREQ, RD und WAIT (S1- 6,5,1) zu schließen, die HEX-Adreß-Schalter werden auf die Adresse '0 8 4 4' gestellt und die Adreß-Komparatoren über die DIL-Schalter S2-1,2,3,4 aktiviert. Der Computer hält dann während des entsprechenden Speicherzugriffs an.

Jetzt werden die DIL-Schalter S2- 1,2,3,4 geöffnet und der Schalter S2- 5 für die Steuersignale geschlossen. Als Steuersignale verwendet man weiterhin MREQ und RD. Nach Drücken der Digi-Taste (T1) übergeht die CPU den Speicher-Refresh, und es erfolgt der zweite Opcode Fetch (Byte 5B).

Den Speicher-Refresh übergeht man wieder durch Betätigen der Taste T1. Sinngemäß erfolgt dann zweimal ein Memory-Read-Zyklus (Bytes DC,0E). Diese Speicherzugriffe kann die Karte eventuell nicht mehr auflösen (siehe Single-Step-Betrieb), wenn man den Vorgang vom zweiten Opcode Fetch neu gestartet hat. Die Zugriffe können aber ausgelöst werden, wenn auf diesem Speicherzugriff ein WAIT-Zustand liegt.

Der Befehl ist jetzt komplett gelesen. Die CPU führt zwei weitere Memory-Read-Zyklen durch, in denen sie die Bytes aus den Speicherzellen 3322h und 3323h in das DE-Register lädt.

Will man nur wissen, welchen Inhalt die CPU aus den Speicherzellen 0EDCh und 0EDDh in das DE-Register liest, kann man den ECB-Bus-Monitor auf die Adresse 0EDCh einstellen und auf die Signale MREQ und RD 'triggern'. Nachdem die CPU das erste Byte gelesen hat, wird es über die Leuchtdioden LD1 bis LD8 angezeigt. Das zweite Byte kann bei einem zweiten Programmdurchlauf zur Anzeige gebracht werden, wenn die Adresse 0EDDh eingestellt ist.

Hat man jedoch den Programmlauf bei der Adresse 0EDCh über WAIT gestoppt, stellt man die Adresse 0EDDh ein und beendet den WAIT-Zustand durch Betätigen des Tasters T1. Vor dem nächsten Memory-Read-Zyklus (Adresse 0EDDh) wird das Datenregister und somit die Anzeige mit dem Inhalt aus der Adresse 0EDCh aufdatiert.

Ist die WAIT-Logik weiterhin aktiviert, wird der Inhalt aus der Speicherzelle 0EDDh erst beim Abschluß des WAIT-Zyklus am Ende des Memory-Read-Zyklus angezeigt.

DIL-Schalter S1	DIL-Schalter S2
1: WAIT-Logik	A15 bis A12
2: ———	A11 bis A8
3: WR	A7 bis A4
4: MREQ	A3 bis A0
5: RD	verknüpfte Bus-Signale
6: RFSH	—————
7: IORQ	—————
8: MI	—————

Bei geschlossenem Schalter sind die entsprechenden Signale aktiviert.

HEX-Schalter	Adreß-Bits
	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
HEX 1	x x x x
HEX 2	x x x x
HEX 3	x x x x
HEX 4	x x x x

Tabelle 1. Funktionen der Bedienelemente des Bus-Monitors

LD1...LD8	Daten D0 bis D8
LD9	Datenwort ist übernommen
LD10	A0 bis A3 gleich gewählter Adresse
LD11	A4 bis A7 gleich gewählter Adresse
LD12	A8 bis A11 gleich gewählter Adresse
LD13	A12 bis A15 gleich gewählter Adresse
LD14	WAIT-Logik aktiv

Tabelle 2. Funktionen der Leuchtdioden

Ein anderes Beispiel: Die Funktion von Software-Timern ist in der Praxis nicht einfach zu prüfen. Arbeitet man mit einem Debugger (DDT, ZSID, TRACE80), sind die Timer meistens durch wiederholten Interrupt abgelaufen, ehe man ihre Funktion prüfen konnte. Man kann den Timer nach Programm 1 mit der Monitor-Karte überprüfen. Der Software-Timer (Start bei Adresse 160h) wird 'rückwärts' gezählt. Der Start des Timers erfolgt nach der Eingabe eines beliebigen Zeichens über die Tastatur. Das Zeichen wird in der Speicherzelle (CHAR) vorübergehend gespeichert. Ist der Timer abgelaufen, so wird dieses Zeichen gelöscht.

Um die Funktion des Timers mit dem Bus-Monitor zu beobachten, sind folgende Einstellungen vorzunehmen: Die DIL-Schalter S1-3, 4 und S2-1, 4 sind auf ON zu stellen. Mit den HEX-Schaltern wählt man die Adresse 160h (0-1-6-0). Will

man nur beobachten, was in der Speicherzelle (CHAR) geschieht, ist die Adresse 161h (0-1-6-1) einzustellen.

Erfahrungen

Bei der Entwicklung eines Systems kann die Monitor-Karte ebenfalls gute Dienste leisten. Wer kennt nicht die Situation, daß ein Computer plötzlich 'hängt'. Hier hilft es oft weiter, wenn man (im Betrieb) den ECB-Bus-Monitor zusteckt und so zumindest die Stelle im Programm finden kann, an der der Computer 'kreist' und auf besondere Bedingungen wartet. Durch Verstellen der HEX-Schalter kann man leicht die zugehörigen Adressen finden. Die Leuchtdioden LED1 bis LED8 leuchten dann, und die LED 9 zeigt an, daß alle vier Adreßteile gleichzeitig aktiviert sind.

Der Anschluß der Monitor-Karte an ein ECB-Bus-System dürfte kaum Probleme berei-

ten. Man hat lediglich dafür zu sorgen, daß man die Bedienele-

Adaptiert

mente der Monitor-Karte bequem erreichen kann. Die Karte läßt sich aber auch über einen Adapter mit dem Bus verbinden. Eine andere Anschlußmöglichkeit wäre, die Karte zu teilen und die beiden Hälften über ein kurzes Stück Flachbandkabel zu verbinden.

Statt CPU

Zum Anschluß des Bus-Monitors an Einplatinen-Computer den CPU-Adapter muß man sich einen speziellen CPU-Adapter anfertigen. Dazu benötigt man einen 40poligen DIL-Stecker ohne Gehäuse, an

den ein 40poliges Kabel angelötet wird, das nicht länger als 35 cm sein sollte. Auf die nach oben überstehenden Lötstifte des Steckers lötet man eine 40polige IC-Fassung, in die eine Z80A-CPU eingesetzt wird. Das andere Ende des Kabels ist mit einer 64poligen VG-Buchsenleiste nach Tabelle 3 zu verlöten. Die vom Monitor nicht verwendeten Leitungen brauchen nicht angeschlossen zu werden.

Für den Testvorgang setzt man für die CPU im Einplatinen-Computer den CPU-Adapter ein und verbindet die ECB-Bus-Monitor-Karte mit dem Adapter. Bei einer Kabellänge von 35 cm und einer CPU-Taktfrequenz von 2,5 MHz ist der Be-

(0100) ORG 0100H			
0100	2A0100	START: LD	HL, (00001H) ; ADRESSE BIOS/WARMBOOT
0103	23	INC	HL
0104	23	INC	HL
0105	23	INC	HL ; ADRESSE BIOS/CONST
0106	111B01	LD	DE, CONST
0107	010700	LD	BC, 00009
010C	ED00	LDTR	; JP-TAB. ANLEGEN
010E	3E00	LD	A, 000H
0110	326001	LD	(TIMER), A
0113	3E00	LD	A, 000H
0115	326101	LD	(CHAR), A
0118	C32401	JP	MAINLP
;			
011B	C30000	CONST: JP	00000H ; CONSOLE STATUS
011E	C30000	CONIN: JP	00000H ; CONSOLE TASTATUR
0121	C30000	CONOUT: JP	00000H ; CONSOLE BILDSCHIRM
;			
0124	CD1B01	MAINLP: CALL	CONST ; ZEICHEN VORHANDEN ?
0127	B7	OR	A
0128	2010	JR	Z, MAIN1 ; NEIN
;			
012A	CD1E01	CALL	CONIN ; ZEICHEN HOLEN
012D	326101	LD	(CHAR), A
0130	4F	LD	C, A
0131	CD2101	CALL	CONOUT
0134	3A5F01	LD	A, (ZK8)
0137	326001	LD	(TIMER), A
;			
013A	2A5D01	MAIN1: LD	HL, (ZK16) ; ZEITKONSTANTE
013D	2B	MAIN2: DEC	HL ; DELAY SCHLEIFE
013E	7C	LD	A, H
013F	B5	OR	L
0140	20FB	JR	NZ, MAIN2
;			
0142	3A6001	MAIN3: LD	A, (TIMER)
0145	B7	OR	A
0146	200B	JR	Z, MAIN4
0148	3D	DEC	A
0149	326001	LD	(TIMER), A
014C	2005	JR	NZ, MAIN4
014E	3E00	LD	A, 000H
0150	326101	LD	(CHAR), A ; (CHAR): 00
;			
0153	3A6101	MAIN4: LD	A, (CHAR)
0156	FE2E	CP	; ENDE-ZEICHEN ?
0158	20CA	JR	NZ, MAINLP
015A	C30000	JP	00000H ; WARM-BOOT
;			
015D	C43B	ZK16: DEFW	15300 ; 100 MILLISEC. / 4 MHZ
015F	14	ZK8: DEFB	20 ; 20 x 100 MILLISEC.
;			
0160	(0001)	TIMER: DEFS	1
0161	(0001)	CHAR: DEFS	1

Programm 1. Eine Test-Routine für den Bus-Monitor

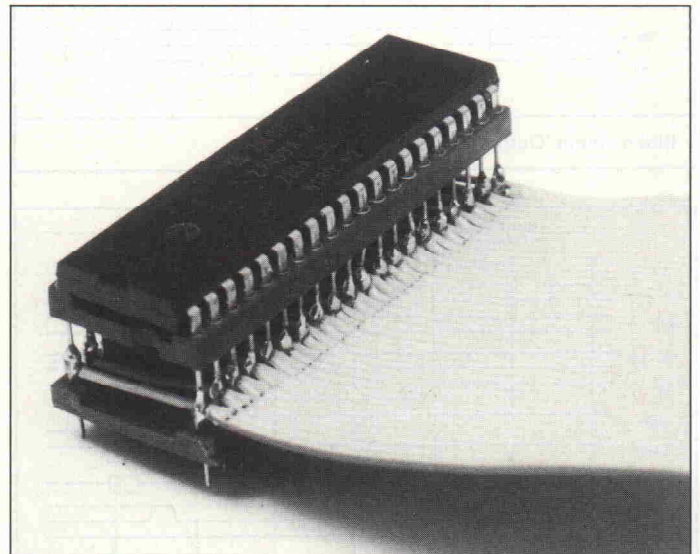


Bild 9. CPU-Adapter, bestehend aus einem DIL-Stecker, IC-Fassung, Z80A-CPU und Kabel.

Ader-	CPU-	Signal	VG-	Ader-	CPU-	Signal	VG-
Nr.	Pin		Feder	Nr.	Pin		Feder
1	1	A11	17 c	21	11	+5V	1 a, c
2	40	A10	18 c	22	30	A 0	5 c
3	2	A12	27 c	23	12	D 2	4 c
4	39	A 9	30 a	24	29	GND	32 a, c
5	3	A13	29 a	25	13	D 7	3 c
6	38	A 8	8 c	26	28	RFSH	28 a
7	4	A14	18 a	27	14	D 0	2 c
8	37	A 7	9 c	28	27	M 1	20 a
9	5	A15	28 c	29	15	D 1	14 c
10	36	A 6	9 a	30	26	RESET	26 c
11	6	CLOCK	29 c *	31	16	INT	21 c *
12	35	A 5	8 a	32	25	BUSRQ	11 a *
13	7	D 4	5 a	33	17	NMI	20 c *
14	34	A 4	7 a	34	24	WAIT	10 a
15	8	D 3	4 a	35	18	HALT	25 c *
16	33	A 3	6 c	36	23	BUSAK	31 a *
17	9	D 5	2 a	37	19	MREQ	30 c
18	32	A 2	6 a	38	22	WR	22 c
19	10	D 6	3 a	39	20	TORQ	27 a
20	31	A 1	7 c	40	21	RD	24 c

* Diese Leitungen sind auf der Karte nicht benutzt

Tabelle 3. Leitungsaufteilung beim CPU-Adapter

trieb an den ungepufferten CPU-Leitungen in fast allen Fällen möglich. Man muß allerdings berücksichtigen, daß die Belastung der CPU-Anschlüsse um eine LS-TTL-Last (ent-

spricht 1/3 einer 'normalen' TTL-Last) vergrößert wird. Die meisten CPUs können aber größere Lasten treiben, als in den Spezifikationen angegeben wird. □

Stückliste

Widerstände

R14...R21, 270R
R43...R48 330R
R1 1k0
R22...R37 2k2
R2...R5 4k7
R6...R13 10k
R49, R50 27k
R38...R42

Kondensatoren

C1...C5 22µF/16V, Tantal
C6 33pF (siehe Text)
C7 180...560pF
CB 6 Stützkondensatoren je 100nF, RM 5 mm

Halbleiter

IC1 74LS273
IC2...IC5 74LS85
IC6 74LS05
IC7 74LS30
IC8...IC10 74LS123
IC11 74LS01
IC12, IC13 ULN2004

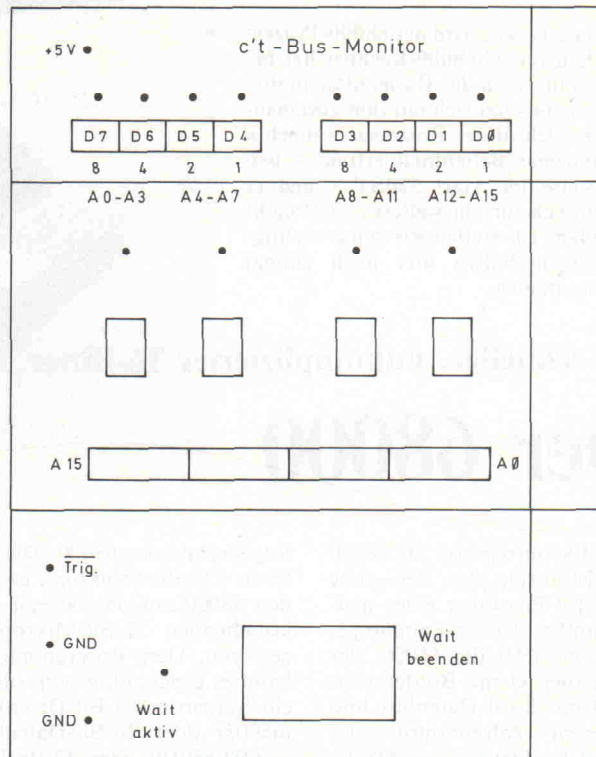
LED1...

LED14 Leuchtdioden, 2 bis 5 mm
D1 AA119

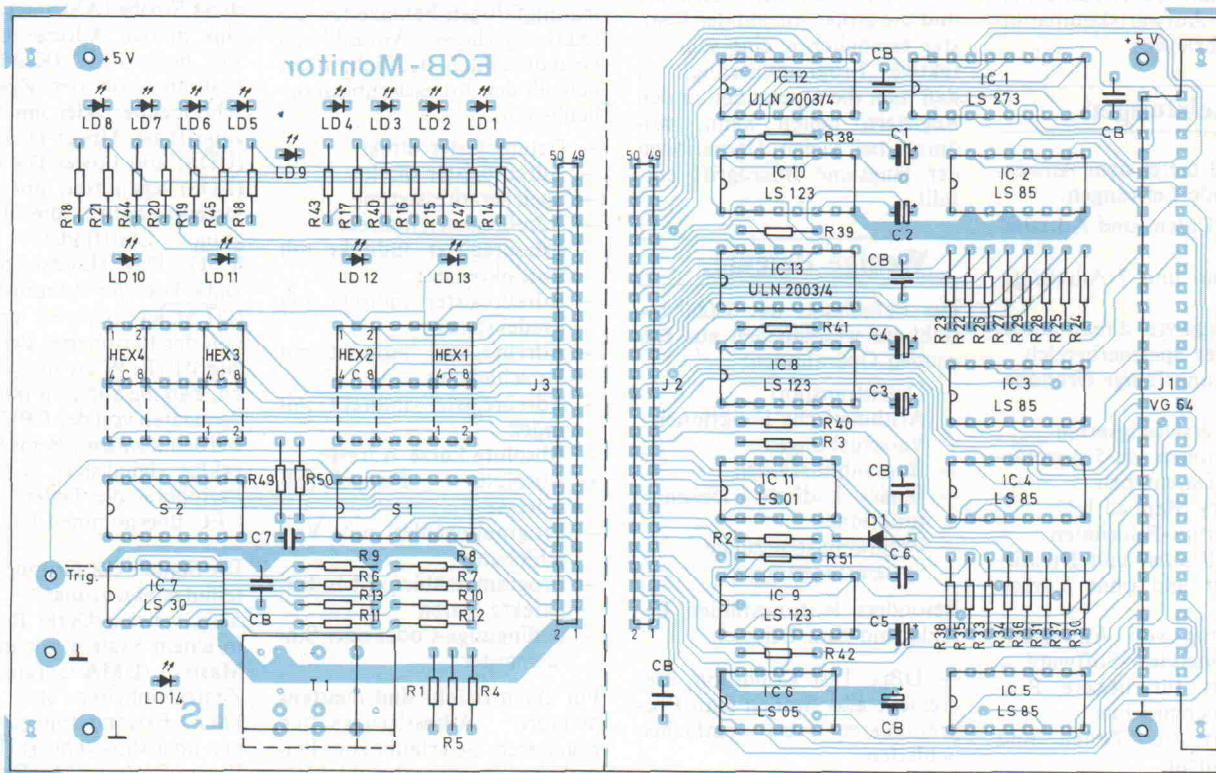
Sonstiges

S1, S2 DIL-Schalter 8-polig
HEX1... HEX4 HEX-Schalter, Firma KEL/Japan
T1 Digi-Taster, Firma Shadow oder Seuffer
J1 VG-Steckerleiste, Reihen a und c bestückt, 64polig
Pfohlenleisten 50polig, siehe Text
J2, J3

IC-Fassungen: 3 x 14polig, 13 x 16polig, 1 x 20polig; 2 Schrauben M 2,5 x 10 mit Muttern, 4 Löt Nägel 1,4 mm, Platine 'c't-Bus-Monitor'.



Eine Schablone, die die Funktion der Bedienteile verdeutlicht. Wer es wünscht, kann sie ausschneiden, auf Pappe kleben und als 'Frontplatte' für den Monitor verwenden.



Der Bestückungsplan. Je nach Ausführung der HEX-Schalter gilt die durchgehend oder gestrichelt gezeichnete Position.