Sonderheft Nr. 204
Preis 28 DM
28 sfr., 210 öS
mc

Die Serie wurde vom Bundesministerium für Forschung und Technologie gefördert

MIKRO-COMPUTER 2

Schritt für Schritt

Der Vollausbau des NDR-Computers

Z80 und CP/M: Von der Hardware zum Betriebssystem

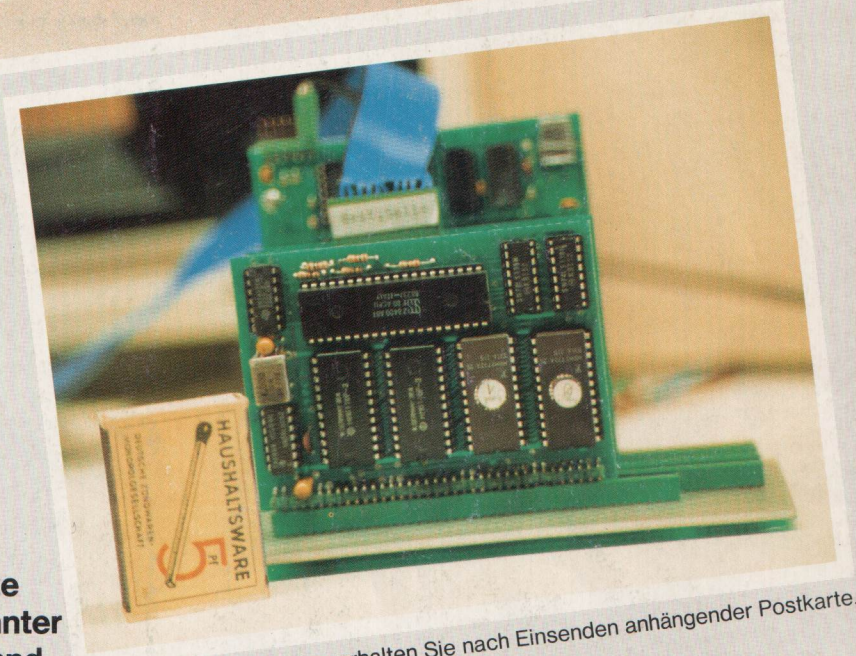
68008 und CP/M: Vom Maschinenbefehl bis zum Floppy-Betrieb alles verständlich

„Mikroelektronik"
Das zweite Sonderheft zur Fernsehserie

Wenn Sie wirklich wissen wollen, wie ein Computer funktioniert – bauen Sie ihn doch einfach selbst!

Der NDR-Computer

vom günstigen Einsteigerpaket, auch für Anfänger, bis zum professionellen System für Profis.



Die Bausätze und Fertiggeräte erhalten Sie bei uns in gewohnter Spitzenqualität aus erster Hand.

Mehr Information erhalten Sie nach Einsenden anhängender Postkarte.

Graf Elektronik Systeme GmbH, Postfach 1610, 8960 Kempten, Tel. (08 31) 62 11

Filiale Hamburg, Ehrenbergstr. 56, 2000 Hamburg 50 (Altona), Telefon (0 40) 38 81 51
Filiale München, Georgenstr. 61, 8000 München 40 (Schwabing), Telefon (0 89) 2 71 58 58

2800 Bremen, GMCP GmbH, Werftstr. 160, 2800 Bremen 21, Tel. (04 21) 61 30 15
A-1030 Wien, ISYS-Informationssysteme, Landstraßer Hauptstraße 2a, Tel. 75 33 25 CH-4106 Therwil, SYSTECH, Postfach, Tel. (0 61) 73 49 73

ges

Dr. Hans Hehl

Der Grafikprozessor EF9366

Beim NDR-Klein-Computer wird eine sogenannte GDP64K-Baugruppe verwendet. Ihr Aufbau und der Funktionstest wurden im Sonderheft Nr. 88, „Mikrocomputer Schritt für Schritt, Teil 1", beschrieben. Nun soll der dabei verwendete Grafikprozessor EF9366 der Firma Thomson etwas genauer untersucht werden.

Die drei Buchstaben G, D, P sind die Anfangsbuchstaben der drei Wörter „GRAPHIC DISPLAY PROCESSOR". Die Zahl 64 gibt die Speicherkapazität (64 KByte) des Bildwiederholspeichers an.

Im Prinzip ist die Karte ein eigenständiger Mikrocomputer mit eigenem Hauptspeicher. Vom GDP werden vor allem die zur Ansteuerung eines Monitors notwendigen Signale erzeugt, also das Vertikal- und das Horizontal-Synchronsignal, sowie das Videosignal, die Bildinformation. Diese wird aus dem Hauptspeicher (Bildwiederholspeicher) entnommen, wobei einem dunklen Bildpunkt ein gesetztes Bit entspricht. Genauere Einzelheiten über die Erzeugung eines Monitorbildes sind im Buch „Mikrocomputer selbstgebaut und programmiert" von R.-D. Klein geschildert (auch das Studium des Datenblattes des Prozessors ist letztlich zu empfehlen).

Registeradresse	A3	A2	A1	A0	Registerfunktion
70h	0	0	0	0	Status-Register, nur lesend (R/W = 1) Befehlsregister (CMD), nur schreibend (R/W = 0)
71h	0	0	0	1	1. Kontrollregister, Schreibvorgänge u. Interruptsteuerung
72h	0	0	1	0	2. Kontrollregister, Zeichenstrichart der Vektoren, Orientierung der Buchstaben
73h	0	0	1	1	Register für Zeichengröße in X- u. Y-Achse (C-SIZE)
74h	0	1	0	0	nicht verwendet
75h	0	1	0	1	Delta-X-Register (siehe Vektorbefehle)
76h	0	1	1	0	nicht verwendet
77h	0	1	1	1	Delta-Y-Register (siehe Vektorbefehle)
78h	1	0	0	0	X-Register, Bit 0 - 3 (MSB) X-Koordinaten (256 - 4095)
79h	1	0	0	1	X-Register, Bit 0 - 7 (LSB) X-Koordinaten (0 - 255)
7Ah	1	0	1	0	Y-Register, Bit 0 - 3 (MSB) Y-Koordinate (256 - 4095)
7Bh	1	0	1	1	Y-Register, Bit 0 - 7 (LSB) Y-Koordinate (0 - 255)
7Ch	1	1	0	0	X-Lichtgriffelregister, nur lesend Bit 2-7, Bit 1 immer 0, Bit 0 = Status
7Dh	1	1	0	1	Y-Lichtgriffelregister, nur lesend Bit 0 - 7
7Eh	1	1	1	0	nicht verwendet
7Fh	1	1	1	1	nicht verwendet

Bild 2. Die Register des Prozessors 9366

Was der EF9366 kann

Vektoren werden mit bis zu 1,5 Millionen Bildpunkten pro Sekunde ausgegeben. Ein integrierter Zeichengenerator erzeugt in einer 5×8-Bildpunkte-Matrix die Zeichen des amerikanischen Stan-

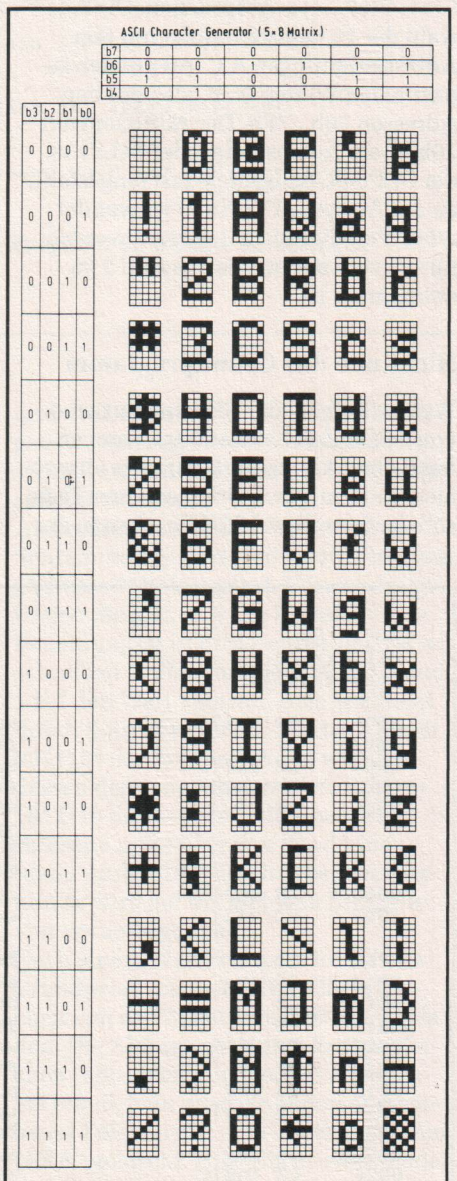


Bild 1. Der ASCII-Zeichensatz, der in dem EF-9366 eingebaut ist

dard-ASCII-Zeichensatzes aus den binär codierten Zahlen von 20 bis 127 (*Bild 1*). Leider gibt es keine Umlaute und kein „ß". Die vorhandenen Zeichen sind in fast beliebiger Form auf dem Bildschirm darstellbar.

Alle Register ziehen?

Der Prozessor erhält die dafür notwendigen Informationen über verschiedene Ein- und Ausgabeeinheiten, sogenannte Register. In diese Register können Zahlenwerte von einem Hauptrechner eingeschrieben werden, die dann dort gespeichert und ausgewertet werden. Umgekehrt kann der Prozessor über diese Register Informationen ausgeben.
Der GDP wird beim NDR-Klein-Computer über die Adressenleitungen A7...A4 mit dem sedezimalen Wert 7 (binär 0111, Bit 7...4) angesprochen. Die Auswahl der 16 Register erfolgt mit den Adressenleitungen A3...A0 mit den sedezimalen Werten 0...F, also mit den Adressen 70h...7Fh. Der Adressenwert 70h ist am Adressendecoder 74LS138 der GDP64K-Baugruppe fest eingestellt, da der Ausgang Y7 (Pin 7) verwendet wird. Von diesen 16 Registern werden nur 12 verwendet, wie aus *Bild 2* zu entnehmen ist.

```
gdp:=  70            ;GDP-Registeradresse
se:=   60            ;Seitenportadresse
ci:=   24            ;für SBC2: Zeicheneingabe

8800    CD 882A      call warte      ;warten bis bereit
8803    AF           xor  a          ;Akku löschen
8804    D3 60        out  (se),a     ;Seite 0 anwählen
8806    CD 882A      call warte      ;warten bis fertig
8809    3E 03        ld   a,3h       ;3h = 0011
                                ;Bit 0 gesetzt: Stift unten
                                ;Bit 1 gesetzt: schreiben
880B    D3 71        out  (gdp+1),a ;ausgeben an GDP
880D    CD 882A      call warte      ;warten bis fertig

                     aus:
8810    CD 0024      call ci         ;Zeichen von Tastatur
8813    FE 1B        cp   1bh        ;Escape-Taste ?
8815    C8           ret  z          ;dann zurück zum Menue
8816    F5           push af         ;Akku-Wert retten
8817    CD 8822      call era        ;jetzt altes Zeichen löschen
881A    CD 882A      call warte      ;warten bis fertig
881D    F1           pop  af         ;alten Akku-Wert regenieren
881E    D3 70        out (gdp),a     ;neues Zeichen an GDP
8820    18 EE        jr   aus        ;nochmal das Ganze

                     era:
8822    CD 882A      call warte      ;warten bis bereit
8825    3E 06        ld   a,6h       ;6h = 0110b, Bit 1 u. 2 gesetzt
                                     ;X-, Y-Register u. Bildschirm
                                     ;löschen
8827    D3 70        out  (gdp),a    ;Befehl an CMD-Register
8829    C9           ret             ;zurück

                     warte:
882A    DB 70        in   a,(gdp)    ;Status abfragen, ob fertig
882C    E6 04        and  4h         ;4h = 0100b, Maske Bit 2
882E    28 FA        jr   z,warte    ;Bit 2 = 0, nochmal warten
8830    C9           ret             ;zurück
```

Bild 4. Dieses Z80-Programm gibt ebenfalls ASCII-Zeichen aus

Nicht nur das Grundprogramm

Welche Befehle der GDP zur Ausgabe von ASCII-Zeichen benötigt, kann anhand eines kleinen Programmes ausprobiert werden. Nach Drücken einer Taste soll das jeweilige ASCII-Zeichen in der linken unteren Bildschirmecke erscheinen.

Wenn Sie diese Aufgabe mit einem Programm lösen wollen, dann können Sie einige Befehle des Grundprogrammes verwenden, wie in *Bild 3* gezeigt. *Bild 4* zeigt ein Programm, bei dem der GDP ohne Befehle aus dem Grundprogramm angesteuert wird. Dabei ist zu beachten, daß nur der Befehl CD 0024 (Adresse 8810) als CD 0024.W einzugeben ist. Auch unter CP/M laufen die Beispielprogramme, wenn kleine Anpassungen vorgenommen werden. Das Programm muß hier bei der Adresse 100h beginnen. *Bild 5* enthält hierzu das Listing des gleichen Programmes als Assembleraus-druck. Ebenso geht es mit dem 8-KByte-Basic (in zwei EPROMs auf SBC2), wobei der Ausgabebefehl OUT (Portadresse, Wert) und der Befehl WAIT (Portadresse, Y, Z) für die Abfrage des Statusregisters des GDP Verwendung finden. *Bild 6* zeigt das dazugehörige Basic-Programm. Diese Beispielprogramme sind möglichst einfach gehalten und sollen zum Experimentieren anregen.

```
GDP:= 70
SEITE:= 60
Taste:= 8810

8800    CD WAIT          warten bis GDP bereit
8803    AF               Akku löschen
8804    D3 SEITE         Seite 0 anwählen
8806    CD WAIT          warten bis GDP bereit
8809    3E 03            Akku mit 3 laden
                         3h = 0011
                         Bit 0 gesetzt: Stift unten
                         Bit 1 gesetzt: schreiben
880B    D3 GDP+1         ausgeben an GDP
880D    CD WAIT          warten bis GDP fertig

8810    CD CI            Zeichen von Tastatur
8813    FE 1B            Escape-Taste gedrückt?
8815    C8               dann zurück zum Menue
8816    F5               Zeichen auf Stapel retten
8817    CD WAIT          warten bis GDP fertig
881A    3E 06            Akku mit 6 laden (löschen; X,Y = 0)
881C    D3 GDP           ausgeben an GDP
881E    CD WAIT          warten bis GDP fertig
8821    F1               Zeichen wieder holen
8822    D3 GDP           an GDP ausgeben
8824    C3 TASTE         zurück zu 8810
```

Bild 3. So können unter dem Z80-Grundprogramm ASCII-Zeichen ausgegeben werden

Neue Bekanntschaft

Anhand des Z80-Maschinenprogrammes für die SBC2 (Bild 4) fällt es nicht schwer, den GDP kennenzulernen. Die einzelnen Programmschritte lassen sich

mit dem Grundprogramm-Menü <3 = Einzelschritt> verfolgen.
Am Programmbeginn empfiehlt es sich, das Statusport 70h abzufragen, ob der GDP bereit ist. Dies geschieht in einem Unterprogramm „warte" mit einer Abfrageschleife, bei der mit der Bitmaske 0100 durch eine UND-Verknüpfung mit dem Akku-Wert der Z80-CPU so lange gewartet wird, bis Bit 2 des Statusports den Wert 1 annimmt. Dann erst kann der GDP neue Befehle entgegennehmen. Mit dem Befehl JR Z,WARTE erfolgt der Rücksprung zum Unterprogrammanfang.
Zu Beginn wird durch Beschreiben des Seitenports (Adresse 60h) mit dem Wert 0 die Schreib- und Leseseite 0 eingestellt. Insgesamt gibt es vier Bildseiten zu je 512 × 256 Bildpunkten. Ausgewählt wird die Schreibseite mit Bit 7 und 6 und die Leseseite mit Bit 5 und 4 des an dem Seitenport ausgegebenen sedezimalen Wertes. Bei zwei Bits sind vier Möglichkeiten vorhanden, nämlich die Bitkombinationen 00, 01, 10 und 11, die den Bildseiten 0, 1, 2 und 3 entsprechen. Es kann eine Bildseite beschrieben und gleichzeitig eine andere angezeigt werden. Der sedezimale Wert 70 (01110000) ergibt die Schreibseite Nr. 1 und die Leseseite Nr. 3. Zur gleichzeitigen Darstellung zweier Bildseiten muß schnell zwischen diesen hin- und hergeschaltet werden.
Ab Programm-Adresse 8809 wird der Schreibmodus im ersten Kontrollregister (Adresse 71h) festgelegt. Verwendung finden die Bits 0...6 des Registers. Ihre Bedeutung ist in *Bild 7* enthalten. Da wir schreiben wollen, müssen wir Bit 0 und 1 auf den Wert 1 setzen. Dies geschieht durch Einschreiben des Wertes 3 (binär 0011) in das Register. Wenn wir statt dessen den sedezimalen Wert B (binär 1011) in das Register schreiben, kann die Bildseite nicht über den Rand hinaus (dann unsichtbar) beschrieben werden und es wird auf der gleichen Bildzeile von vorne begonnen.
Nach einem Wartezyklus wird mittels Grundprogrammbefehl CD CI ein Zeichen von der Tastatur in den Akku geladen. Die Abfrage nach dem Escape-Zeichen (1Bh) ermöglicht einen Ausstieg ins Menü. Nun wird ein schon ausgegebenes Zeichen mit dem Unterprogramm „ERA" gelöscht. Hier gibt es verschiedene Möglichkeiten, die sich durch Austauschen des Werts an der Adresse 8826 ausprobieren lassen. *Bild 8* zeigt die Bit-

```
 10 CLRS
 20 GDP = HEX("70")
 30 INPUT"eine Taste u. CR druecken";A$
 35 CLRS
 40 IF ASC(A$) = 27 THEN 100          :REM Escape beendet
 50 PAGE 0,0                          :REM Seite 0
 55 MOVETO 0,0                        :REM linke, untere Ecke
 60 OUT GDP, ASC(A$)                  :REM 1.Zeichen an GDP
 70 WAIT GDP,4,0                      :REM warten bis Bit 2 = 1
 80 POKE HEX("87C5"),0                :REM Cursor aus
 90 GOTO 30
100 END
```

Bild 6. Das Programm in Basic für die Platine SBC2

```
Bit 0       1 : schreiben          0 : nicht schreiben

Bit 1       1 : nicht löschen      0 : löschen

Bit 2       1 : Schnell schreiben ohne Bildschirmausgabe

Bit 3       1 : nur im Bildfenster schreiben
            0 : Bildfenster = 4096 * 4096 Bildpunkte

Bit 4       1 : Interrupt-Freigabe nach einer Lichtgriffel-
                sequenz

Bit 5       1 : Interrupt-Freigabe bei jedem Vertikal-
                Synchronsignal

Bit 6       1 : Interrupt-Freigabe, wenn der GDP bereit

Bit 7       0   nicht verwendet
```

Bild 7. Die Bedeutung der Bits im Kontrollregister. Bit 0 und 1 werden auch durch das Befehlsregister verändert

```
gdp   equ   70h         ;GDP-Registeradresse
se    equ   60h         ;Seitenportadresse
ci    equ   0F003H      ;MC-Comp. Zeicheneingabe

0100    CD 012C     call warte      ;warten bis bereit
0103    AF          xor  a          ;Akku löschen
0104    D3 60       out  (se),a     ;Seite 0 anwählen
0106    CD 012C     call warte      ;warten bis fertig
0109    3E 03       ld   a,3h       ;3h = 0011
                                      ;Bit 0 gesetzt: Stift unten
                                      ;Bit 1 gesetzt: schreiben
010B    D3 71       out  (gdp+1),a ;ausgeben an GDP
010D    CD 012C     call warte      ;warten bis fertig

0110                aus:
0110    CD F003     call ci         ;Zeichen von Tastatur
0113    FE 1B       cp   1bh        ;Escape-Taste ?
0115    CA 0000     jp   z,0        ;dann zurück zu CP/M
0118    F5          push af         ;Akku-Wert retten
0119    CD 0124     call era        ;jetzt altes Zeichen löschen
011C    CD 012C     call warte      ;warten bis fertig
011F    F1          pop  af         ;alten Akku-Wert regenieren
0120    D3 70       out  (gdp),a    ;neues Zeichen an GDP ausgeben
0122    18 EC       jr   aus        ;nochmal das Ganze

0124                era:
0124    CD 012C     call warte      ;warten bis fertig
0127    3E 06       ld   a,6h       ;6h = 0110b, Bit 1 u. 2 gesetzt
                                    ;X-, Y-Register u. Bildschirm
                                    ;löschen
0129    D3 70       out  (gdp),a    ;Befehl an CMD-Register
012B    C9          ret             ;zurück

012C                warte:
012C    DB 70       in   a,(gdp)    ;Status abfragen, ob fertig
012E    E6 04       and  4h         ;4h = 0100b, Maske Bit 2
0130    28 FA       jr   z,warte    ;Bit 2 = 0, nochmal warten
0132    C9          ret             ;zurück
```

Bild 5. Dieses Programm gibt die ASCII-Zeichen unter CP/M aus

belegung des CMD-Registers (Adresse 70h). Der binäre Wert 0100 (4) löscht nur den Bildschirm, ohne die Koordinaten der nächsten Schreibposition zu löschen. Der Wert 0101 (5) setzt die Schreibkoordinaten X, Y auf 0, ohne den Bildschirm zu löschen. Schreiben wir den binären Wert 0110 (6) in das Register, so wird der Bildschirm gelöscht und mit dem Schreiben an dem Koordinatenpunkt 0,0 (linke untere Ecke) begonnen. Natürlich muß der Wert des eingegebenen Zeichens mit der Befehlssequenz PUSH AF und POP AF erhalten werden.
Ab Adresse 881E wird dann das Zeichen ausgegeben und das Ganze beginnt bei Adresse 8810 von vorne.

Zeichenzauber oder besondere Wünsche

Im zweiten Kontrollregister wird mit den Bits 2 und 3 die Orientierung der Zeichen verändert, also kann man zum

Befehl Bitfolge 3 2 1 0	Wert	Funktion a) Vektorbefehle b) Buchstaben
a)		
0 0 0 0	0	durchgezogene Linien
0 0 0 1	1	gepunktete Linien - - - - - -
0 0 1 0	2	gestrichelte Linien — — —
0 0 1 1	3	Kombination ——— - ——— - —
b)		
0 0 0 0	0	normale Buchstaben
0 1 0 0	4	schräge Buchstaben
1 0 0 0	8	Buchstaben vertikal
1 1 0 0	C	Buchstaben schräg u. vertikal

Die Bits 4 - 7 sind immer 0

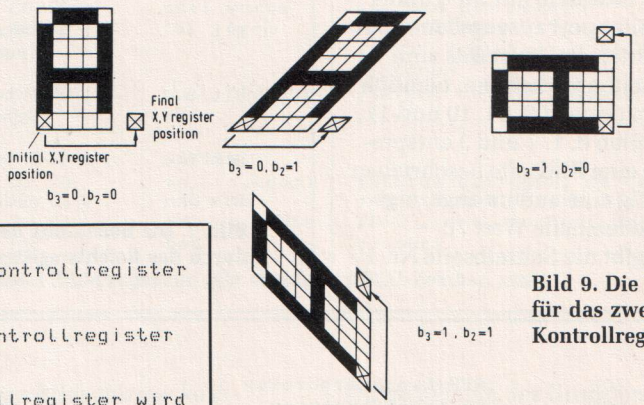


Bild 9. Die Befehle für das zweite Kontrollregister

Bitfolge b3 b2 b1 b0	Wert	Funktion
0 0 0 0	0	Schreibfunktion, im 1.Kontrollregister wird Bit Nr. 1 gesetzt
0 0 0 1	1	Löschfunktion, im 1. Kontrollregister wird Bit Nr. 1 gelöscht
0 0 1 0	2	schreiben, im 1. Kontrollregister wird Bit Nr. 0 gesetzt
0 0 1 1	3	unsichtbar schreiben, im 1. Kontrollregister wird Bit Nr. 0 gelöscht
0 1 0 0	4	Bildschirm löschen
0 1 0 1	5	X- und Y-Register auf 0 setzen
0 1 1 0	6	Bildschirm und X- sowie Y-Register löschen
0 1 1 1	7	Bildschirm und alle Register außer Lichtgriffel löschen, Schriftgröße am kleinsten
1 0 0 0	8	Bildschirm kurz hell für Lichtgriffelposition
1 0 0 1	9	Lichtgriffelfunktion einschalten
1 0 1 0	A	Rechteck ausgeben, Größe P * Q vom Zeichengrößen-Register abhängig
1 0 1 1	B	Quadrat ausgeben, Größe P * Q vom Zeichengrößen-Register abhängig
1 1 0 0	C	Bildschirm hell, Zeichen dunkel
1 1 0 1	D	X-Register löschen
1 1 1 0	E	Y-Register löschen
1 1 1 1	F	Direktzugriff auf Bildschirmspeicher (nur bedingt verwendbar)

Bild 8. Die Befehle für das Befehlsregister (Adresse 70 beim Z80-System)

Beispiel schräg schreiben, wie in *Bild 9* dargestellt. Dazu muß vor einem Schreibbefehl der entsprechende Bitcode sedezimal in das unter Adresse 72h erreichbare Register geschrieben werden.
Damit das Programm nicht ständig neu eingetippt werden muß, können wir an die Adresse 8810 an Stelle der Zeicheneingabe einen Unterprogrammaufruf einfügen. In dem Unterprogramm ab der Adresse 8832 (*Bild 10*) wird dann vor der Zeicheneingabe die Schriftart festgelegt. Außerdem wollen wir auch die Schriftgröße verändern, wobei bis zu 16fache Vergrößerung in der X- und Y-Achse, auch unterschiedlich, möglich ist. Dazu benötigen wir das Zeichengrößen-Register (C-Size) mit der Adresse 73h (*Bild 11*). Die Bits 0...3 enthalten den Vergrößerungsfaktor für die Y-Achse, die Bits 4...7 den der X-Achse. Sedezimale Werte von 1...F ergeben bis zu 15fache Vergrößerung in einer Achse, der Wert 00 ergibt die größten Zeichen (16fach).

```
8810    CD 8832     neues Unterprogramm
 "       "
8832    3E 04       schräge Zeichen
8834    D3 72       Ausgabe an 2. Kontrollregister
8836    CD 882A     call warte
8839    3E 83       Faktor X*8, Y*3
883B    D3 73       Ausgabe an C-Size
883D    CD 882A     call warte
8840    3E 00       X-Koordinate = 0
8842    D3 79       Ausgabe an X-Register LSB
8844    CD 882A     call warte
8847    3E 00       X-Koordinate = 0
8849    D3 78       Ausgabe an X-Register MSB
884B    CD 882A     call warte
884E    3E 64       Y-Koordinate = 100
8850    D3 7B       Ausgabe an Y-Register LSB
8852    CD 882A     call warte
8855    CD 0024     call Zeicheneingabe
8858    C9          ret
```

Bild 10. Ein Z80-Unterprogramm zur Darstellung verschiedener Zeichenformen an verschiedenen Koordinaten

```
 5  CLRS                        :REM Bildschirm löschen
 7  POKE HEX("87C5"),0          :REM Cursor aus
10  GDP = HEX("70")
20  PAGE 0,0                    :REM Bildseite 0
30  OUT GDP+1,11
40  OUT GDP+2,0                 :REM Vektorform
50  MOVETO 200,150              :REM nicht am Rand
                                anfangen
60  KV = INT((255+1-248)*RND(1)+248) :REM F8-FF
70  FOR I = 1 TO 3              :REM Vektorlänge * 3
80  OUT GDP,KV
85  WAIT GDP,4,0                :REM kann entfallen
90  NEXT
100 GOTO 60
```

Bild 12. Ein Programm in Basic, das Kurzvektorausgabe benutzt

```
a) Zeichengrößen-Register (73h)

    Bit 7 - 4      Vergrößerungsfaktor P in X-Achsenrichtung

    Bit 3 - 0      Vergrößerungsfaktor Q in Y-Achsenrichtung

Beispiele:

0001 0001   11   kleinste Zeichengröße:  5 * 8 Punktematrix

0000 0000   00   größte Zeichengröße:   16*5 + 16*8   "

1111 1111   FF   Zeichengröße:          15*5 + 15*8   "

0100 1101   4D       "        :          4*5 + 13*8   "
```

```
b) Kurzvektorbefehle für Befehls-Register (70h)

Bit 7                  immer gesetzt = 1

Bit 6-5 = Bit 4-3:  Vektorlänge in Bildpunkten:

0 0                 :  0 Punkte

0 1                 :  1 Bildpunkt

1 0                 :  2 Bildpunkte

1 1                 :  3 Bildpunkte

Bit 2-0 : Richtungsangabe (gleicher Code wie bei
          Vektorbefehle Gruppe C)

Beispiel für eine Vektorlänge von 3 Bildpunkten:

Richtung                    Befehl
                          dez.  sedez.   binär

 3 Uhr                    248   F8       1111 1000

 1 Uhr 30'                249   F9       1111 1001

12 Uhr                    250   FA       1111 1010

10 Uhr 30'                251   FB       1111 1011

 6 Uhr                    252   FC       1111 1100

 4 Uhr 30'                253   FD       1111 1101

 9 Uhr                    254   FE       1111 1110

 7 Uhr 30'                255   FF       1111 1111
```

Bild 11. Das Zeichengrößenregister (a) und die Kurzvektorbefehle (b)

Letztlich soll der Buchstabe an einer bestimmten Bildschirmstelle erscheinen. Dazu verwenden wir das X- und Y-Register. Dieses wird zwar automatisch bei jeder Zeichenausgabe vom Prozessor verändert, kann aber über die Adressen 78h...7Bh (Bild 2) gelesen und korrigiert werden. Die Koordinaten werden in 12 Bits übergeben, damit sind prinzipiell 4096 × 4096 Bildpunkte ansprechbar. Für eine Bildschirmseite mit 256 Bildpunkten (Y-Achse) × 512 Bildpunkten (X-Achse) brauchen wir für die Y-Koordinate nur das niederwertige Byte (LSB). Der Wert FF (sedezimal) ergibt den Bildpunkt in der linken oberen Bildschirmecke. Soll dort der kleinstmögliche Buchstabe stehen, müssen wir natürlich die Buchstabenhöhe mit 8 Bildpunkten von FF abziehen, das ergibt den Wert F7. Nach dem Eingeben des Ergänzungsprogrammes erscheinen trotz Verwendung der X- und Y-Register die Buchstaben noch in der linken unteren Ecke. Warum?

Sehen Sie sich das Unterprogramm „era" an. Der Löschbefehl besitzt den Wert 6 und dabei wird das X- und Y-Register auf 0 gesetzt. An der Adresse 8826 muß daher der Wert 4 stehen, dann erscheinen die Zeichen an der Koordinate X,Y = 0,100. Verwenden wir bei Adresse 8841 den Wert C0h, dann erscheinen unsere Zeichen etwa in der Bildschirmmitte. Größere Werte als FF können für die X-Koordinate jetzt nicht verwendet werden. Da benötigen wir das höherwertige Byte (MSB) des X-Registers, das den Wert 1 enthalten muß. Das kleinste Zeichen erscheint dann am rechten Bildrand, wenn wir FAh in das LSB und den Wert 1 in das MSB des X-Registers schreiben. Hierzu muß aber an der Adresse 8833 der Wert 0 und an

883A der sedezimale Wert 11 stehen, sonst reicht das Zeichen über den rechten Bildschirmrand hinaus und ist nur zum Teil sichtbar. Die obere rechte Bildschirmecke besitzt die X,Y-Koordinaten 511,255.

Vektoren über Vektoren

Die besonderen Eigenschaften des Grafik-Prozessors kommen erst bei der Vektordarstellung zum Vorschein. Es gibt wiederum sehr viele Möglichkeiten, wobei wir wieder bei den einfacheren, mit den Kurzvektorbefehlen beginnen wollen. Möglich sind eine Längenangabe (1...3 Bildpunkte), die Strichart, z. B. unterbrochen und acht Richtungen. Bild 11 zeigt auch eine Befehlsübersicht zu den Kurzvektorbefehlen.
Zur Angabe der Kurzvektorbefehle dient wieder das Befehlsregister (CMD) mit der Adresse 70h. Verwendung finden die Bytes 80h...FFh, also mit gesetztem Bit 7. Die Register Delta-X und Delta-Y werden dabei nicht benötigt. Die Bits 0...2 codieren die Richtung, in der gezeichnet wird. Die Bits 3...4 bzw. 5...6 enthalten die Bildpunkteanzahl je Kurzvektor. Die Befehlsausgabe mit den sedezimalen Werten F8...FF ergibt also Vektoren mit einer Länge von drei Bildpunkten in acht verschiedene Richtungen. Die Strichart wird durch das zweite Kontrollregister beeinflußt, wobei die Bits 0 und 1 Verwendung finden.
Nachdem so viel mit der Z80-Maschinensprache gearbeitet wurde, sollen die Kurzvektoren hier mit dem 8-KByte-Basic ausprobiert werden. Mit der RND-Funktion erzeugen wir Zufallszahlen zwischen 248 und 255. Diese Zahlen sind als Kurzvektorbefehle interpretierbar. Die Richtung des Kurzvektors ist also zufällig und die mit dem Programm (*Bild 12*) gezeichnete Bahn entspricht in etwa einer zufälligen Teilchenbewegung in einem Gas.
Bild 13 zeigt, wie es nach kurzer Zeit auf dem Bildschirm aussieht. Damit der Vektor im Bereich der sichtbaren Bildschirmseite bleibt, wird in Zeile 30 das Bit 3 des ersten Kontrollregisters zusätzlich zum Schreibbefehl gesetzt. In Zeile 40 wird die Strichart, hier kontinuierlich, an das zweite Kontrollregister mit Adresse 72h ausgegeben. Ein Programmabbruch erfolgt einfach mit der ESCAPE-Taste.

Lange Vektoren

Nun wird es komplizierter. Es sollen Linien von dem Koordinatenpunkt X1,Y1 zum Punkt X2,Y2 gezeichnet werden. Dazu benötigen wir für die Anfangskoordinaten die Register X und Y, für die Differenz der jeweiligen Achsenkoordinaten (entspricht der skalaren Komponente des Vektors bei senkrechter Projektion) die Register Delta-X und Delta-Y sowie das zweite Kontrollregister für die Strichart, zum Beispiel gepunktete Linien, und dann das Befehlsregister. Die sedezimalen Befehlswerte reichen von 10...1F für diese Vektorbefehle.
Wir verwenden wieder ein Basic-Programm (*Bild 14*), da so die Koordinateneingabe kürzer zu programmieren ist, und geben die Befehle direkt an die Register des Prozessors.

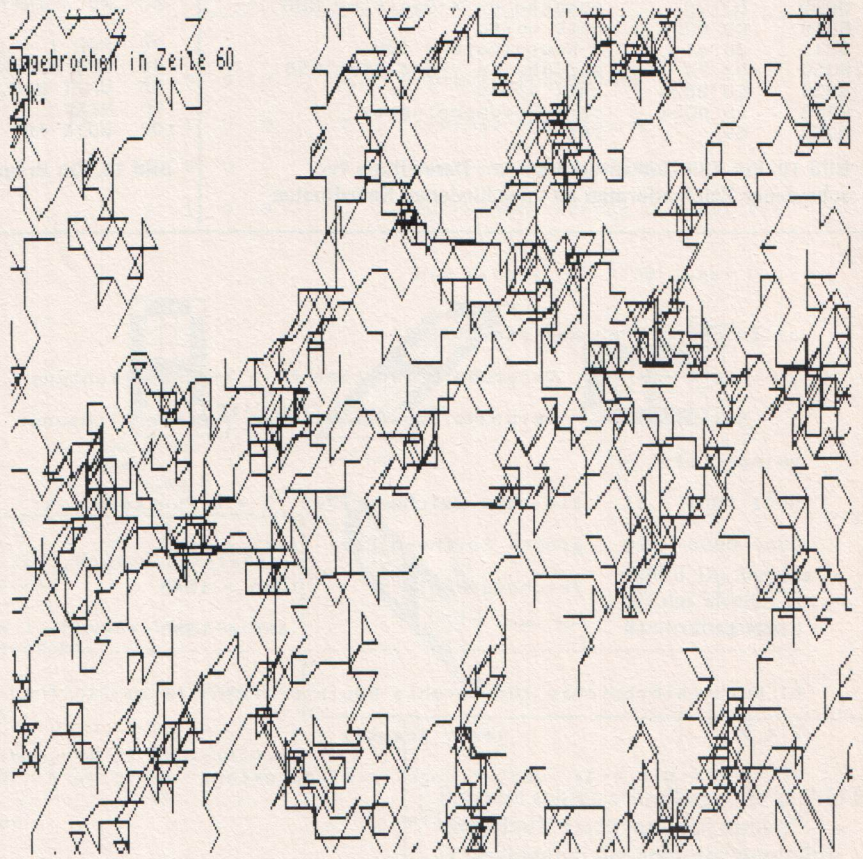


Bild 13. Das zeichnet das Programm aus Bild 12

```
 10 CLRS
 15 POKE HEX("87C5"),0
 20 INPUT"Anfangsadresse X1,Y1";X1,Y1
 30 INPUT"Endadresse     X2,Y2";X2,Y2
 50 GDP = HEX("70")
 60 PAGE 0,0
 70 OUT GDP+1,3           :REM schreiben
 80 OUT GDP+9,X1          :REM X-Register
 90 OUT GDP+11,Y1         :REM Y-Register
100 REM
110 CD = 17               :REM 0001 0111, DX u. DY positiv
120 DX = X2 - X1
130 IF DX<0 THEN CD = 19 :REM 0001 0011, DX negativ
140 ZX = ABS(DX)
150 OUT GDP+5,ZX          :REM Delta X-Register
160 DY = Y2 - Y1
170 IF DY<0 THEN CD = 21 :REM 0001 0101, DY negativ
180 ZY = ABS(DY)
190 OUT GDP+7,ZY          :REM Delta Y-Register
195 REM
200 IF DX<0 AND DY<0 THEN CD = 23 :REM DX u. DY neg.
205 REM
210 WAIT GDP,4,0
220 OUT GDP,CD            :REM Befehlsregister
230 INPUT"NOCHMAL";X$
240 IF X$ = "j" THEN 10
250 END
```

Bild 14. Dieses Programm benutzt Vektorbefehle

Nach Eingabe der Koordinaten in Zeile 20 und 30 wird als erstes das X- und Y-Register mit der Anfangsadresse geladen (Zeile 80 und 90). Falls die Koordinatendifferenz DX bzw. DY negativ wird, muß dies dem Prozessor durch ein gesetztes Bit 1 bzw. Bit 2 im Befehlswort mitgeteilt werden. Daher ergeben sich die vier sedezimalen Befehle 11, 13, 15 und 17, die in *Bild 15*, Gruppe A, enthalten sind. Nach der Absolutwertbildung erfolgt die Ausgabe an die Delta-X- und Delta-Y-Register. Die Zeile 210 kann entfallen, da Basic dem GDP genügend Zeit zur Verarbeitung der Befehle läßt. Mit dem Beispielprogramm kann auf der X-Achse nur der Koordinatenpunkt 255 erreicht werden, da das Delta-Register nur 8 Bit besitzt. Gibt man größere Werte ein, so ergibt sich ein „FC Fehler" (Function call error). Größere Vektoren müssen daher in zwei Teilen gezeichnet werden. *Bild 16* zeigt einen Programmablauf. Mit einer Basic-Zeile, z. B. 75 OUT GDP+2,Wert, kann man entsprechend den Befehlen des zweiten Kontrollregisters die Linienart auswählen.

Die sedezimalen Befehle 10, 12, 14 und 16 (Gruppe B) werden verwendet, wenn das Delta-X-Register oder Delta-Y-Register nicht benötigt werden. Somit ist nur das Vorzeichen des Differenzwertes eines Registers zu berücksichtigen und dementsprechend können die Bits 1 und 2 gesetzt werden.

Noch einfacher sind Vektoren parallel zur Koordinatenachse oder zu Diagonalen mit einer Befehlsgruppe zu zeichnen, die nur das Delta-Register mit der größeren Vektorkomponente auswertet. Entsprechend den acht Richtungsmöglichkeiten gibt es die sedezimalen Befehle 18...1F, die in Bild 15, Gruppe C, gezeigt sind. *Bild 17* zeigt ein Programm zur Ausgabe eines Fadenkreuzes. Die Vektorlänge mit 50 Bildpunkten in der X-Achse wird in das Delta-X-Register geladen. Nach der Berechnung des jeweiligen Startpunktes des Vektors und der Ausgabe in das X- und Y-Register wird der Vektor in die angegebene Richtung gezeichnet.

Der Leser sollte vor allen Dingen den Mut haben, die jeweiligen Parameter der Register in den Beispielprogrammen abzuändern, denn nur durch Experimentieren lernt man den GDP wirklich kennen.

```
Anfangsadresse X1,Y1? 50,40
Endadresse     X2,Y2? 200,75
NOCHMAL?
```

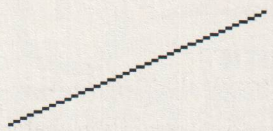


Bild 16. So sieht ein Vektor aus

Vektor-Befehle: Gruppe A

Vorzeichen Delta-X	Vorzeichen Delta-Y	Befehl: dez.	sedez.	binär
+	+	17	11	0001 0001
–	+	19	13	0001 0011
+	–	21	15	0001 0101
–	–	23	17	0001 0111

Vektorbefehle: Gruppe B

Delta-X	Delta-Y	dez.	sedez.	binär
> 0	–	16	10	0001 0000
–	> 0	18	12	0001 0010
–	< 0	20	14	0001 0100
< 0	–	22	16	0001 0110

Vektorbefehle: Gruppe C

Richtung	Befehl dez.	sedez.	binär
3 Uhr	24	18	0001 1000
1 Uhr 30'	25	19	0001 1001
12 Uhr	26	1A	0001 1010
10 Uhr 30'	27	1B	0001 1011
6 Uhr	28	1C	0001 1100
4 Uhr 30'	29	1D	0001 1101
9 Uhr	30	1E	0001 1110
7 Uhr 30'	31	1F	0001 1111

Bild 15. Eine Zusammenstellung der Vektorbefehle

```
 10 REM Fadenkreuz
 20 CLRS
 30 POKE HEX("87C5"),0
 40 INPUT"Koordinate X,Y";X,Y
 50 GDP = HEX("70")
 60 PAGE 0,0
 70 OUT GDP+1,3                              :REM schreiben
 80 GOSUB 300
 90 GOTO 40
100 REM ************************
300 REM AUSGABE
310 VK = 50                                  :REM Strichlänge X-Achse
320 OUT GDP+5,VK                             :REM Delta X-Register
330 KX = X - VK/2: IF KX <0 THEN KX = 0      :REM Startpunkt
340 OUT GDP+9,KX                             :REM X-Register LSB
345 OUT GDP+11,Y                             :REM Y-Register LSB
350 OUT GDP,24                               :REM Vektor nach 3 Uhr
360 WAIT GDP,4,0
390 VK = 25                                  :REM Strichlänge Y-Achse
395 OUT GDP+5,VK                             :REM Delta X-Register
400 KY = Y - VK/2:IF KY<0 THEN KY = 0        :REM Startpunkt
410 OUT GDP+9,X                              :REM X-Register LSB
420 OUT GDP+11,KY                            :REM Y-Register LSB
430 OUT GDP,26                               :REM Vektor nach 12 Uhr
440 RETURN
```

Bild 17. Ein Fadenkreuz mit Vektorbefehlen aufgebaut und ausgegeben