

```

8810  CD 8832  neues Unterprogramm
      .
8832  3E 04    schräge Zeichen
8834  D3 72    Ausgabe an 2. Kontrollregister
8836  CD 882A  call warte
8838  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 FB-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 x 4096 Bildpunkte ansprechbar. Für eine Bildschirmseite mit 256 Bildpunkten (Y-Achse) x 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 COh, 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