

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-Maschinsprache 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 Anfangskordinaten die Register X und Y, für die Differenz der jeweiligen Achsenkordinaten (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 Li-

nien, 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.

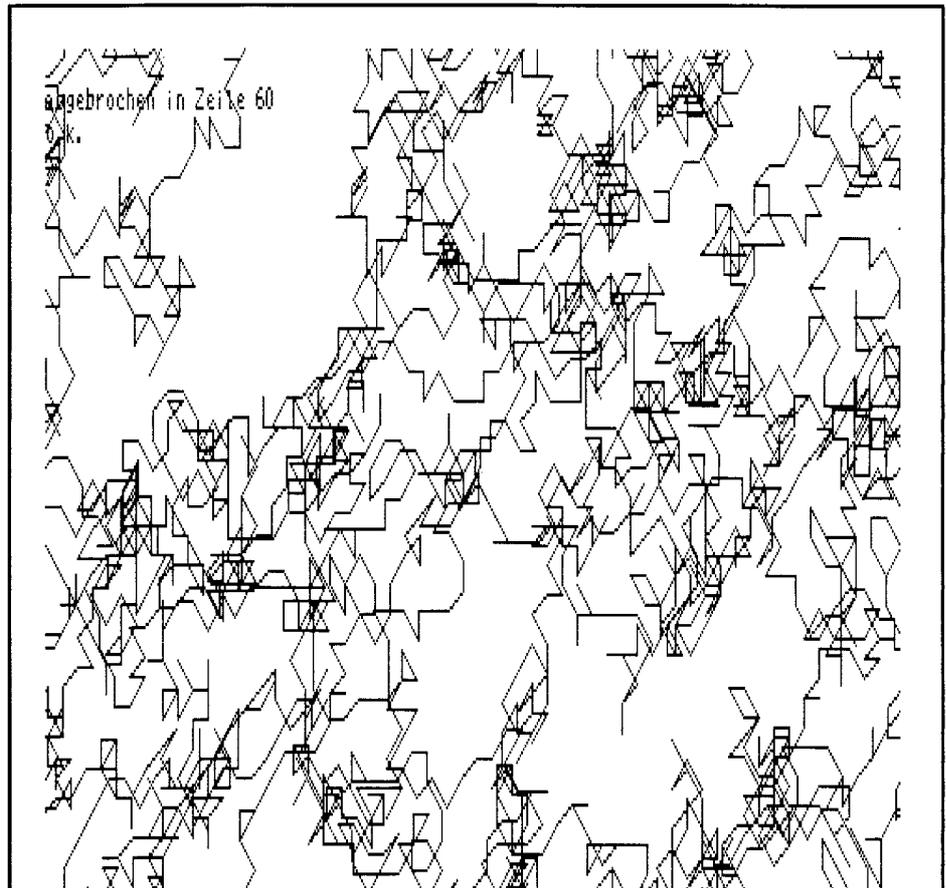


Bild 13. Das zeichnet das Programm aus Bild 12

```

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