

Von Basic zu Assembler

Teil 1

Vom vertrauten Basic zum Programmieren in 6502-Assembler ist es gar nicht so weit. Wir werden langsam anfangen. Zuerst sind die Schleifen dran. Wir werden dabei Basic mit Maschinensprache vergleichen.

Man kann wohl getrost davon ausgehen, daß ein großer Teil der Commodore 64-Benutzer Basic beherrscht. Vermutlich hat es Sie aber auch schon öfters gereizt, die langsamen Programme durch Assembler-Routinen zu beschleunigen. Der Assembler-Kurs »Assembler ist keine Alchimie« hat die Grundlagen dieser Sprache zwar vermittelt, aber bis zum eigenen Maschinenprogramm ist es häufig noch ein holpriger Weg. Ein Führer, der Ihnen den Pfad aus den weiten Basic-Ebenen auf die Gipfel der Assembler-Programmierung zeigt, ist dieser Kurs.

Zunächst halten wir uns an Bekanntes: Basic-Befehle sollen auf ihre Entsprechungen in Assembler untersucht werden. Oft aber streifen wir die Fesseln der Hochsprache ab und lernen allerlei Assembler-Techniken kennen. Wo es uns angemessen erscheint, nehmen wir uns die Freiheit, die computerinterne Firmware zu benutzen. Alles in allem sollen Sie durch Training zum vertrauten Umgang mit Assembler gelangen.

1. Basic contra Assembler

Basic macht uns den Umgang mit unserem Computer relativ leicht: Wir brauchen uns kaum um die Verwaltung von Variablen zu kümmern. Benötigen wir zum Beispiel eine neue Stringvariable, dann genügt ihr erstes Auftauchen und Basic richtet sie für uns gebrauchsfertig ein. Sehr entgegenkommend ist Basic auch, was den Verkehr mit Peripherie jeder Art angeht: Ein Zeichen von der Tastatur anzunehmen, ist mit GET oder INPUT recht einfach. Irgend etwas auf dem Drucker zu schreiben oder auf die Diskette: mit OPEN und dem PRINT # geht's reibungslos und ohne Gedanken an die Busverwaltung. Die Hochsprache nimmt uns vieles ab, worum wir uns in Assembler kümmern müssen.

Andererseits befindet man sich etwa in der Situation eines Bauherrn, der sich ein Haus mit Fertigteilen aufstellen lassen möchte: Jeder Wunsch, der vom Standard abweicht, ist nicht möglich oder wird sehr teuer. Basic-Erweiterungen sind dann mit einer großen Sammlung aller möglichen Fertigteile-Formen zu vergleichen: Man muß sie alle

kaufen, obwohl man nur einige für den Sonderwunsch braucht.

Der Vergleich fängt nun etwas zu hinken an: Die Fertigteile setzen sich nämlich bei genauerem Hinsehen aus kleinen Einzelteilen (Ziegelsteinen) zusammen. Wenn man also auf die vorgefertigten Großteile weitgehend verzichtet, dafür aber den Umgang mit Ziegelsteinen (Assembler-befehlen) beherrscht, kann man sich genau das individuelle Haus bauen, wie bizarr es auch immer aussehen mag.

Wenn man sich das Verarbeiten eines Basic-Programmes etwas genauer ansieht, dann versteht man die Schwerfälligkeit, mit der vieles geschieht. Vom Augenblick des Einschaltens an läuft ein Maschinenprogramm im Computer: Der Interpreter. Jeder Tastendruck, jeder Basic-Befehl, der diesem in die Hände fällt, wird untersucht und führt zur Abarbeitung eines auf diesen Befehl zugeschnittenen Assemblerprogrammes.

Ein Basic-Befehl wie beispielsweise PRINT kann ganz verschiedene Reaktionen erfordern: Wenn vorher ein CMD-Kommando erfolgt war, findet die Ausgabe nicht auf dem Bildschirm statt, wenn das nicht der Fall war, kann der Bildschirm an einer anderen Stelle als im Normalfall liegen. Das, was auszudrucken ist, kann ein String sein, eine Integerzahl oder eine Fließkommavariablen, allerlei Fehlerquellen sind abzufangen etc. Das zu PRINT gehörende Assembler-Programm muß all diese Möglichkeiten berücksichtigen, wie selten sie auch angesprochen werden.

Bei einem individuellen Assembler-Programm wissen wir dagegen, was wir wie ausgeben wollen. Unser eigener »PRINT«-Befehl wird nur das enthalten, was unbedingt notwendig ist, der ganze unnötige Ballast wird von uns nicht programmiert. Allerdings sind wir dann auch voll verantwortlich für die einwandfreie Funktion. Wir müssen beispielsweise dafür sorgen, daß bei Zugriffen auf Register oder Speicher dort der richtige Wert zum richtigen Zeitpunkt im erlaubten Format zur Verfügung steht. Das ist wie im täglichen Leben: Die größere Freiheit legt uns mehr Verantwortung auf und schafft uns andererseits ungeahnte Möglichkeiten, Ideen zu verwirklichen.

2. Noch eine technische Vorbemerkung

Für alle Beispielprogramme werde ich den Assembler Hypra-Ass von Gerd Möllmann verwenden. Er wurde im 64'er, Ausgabe 7/85, Seite 66, als Listing des Monats abgedruckt und ist auch auf der Leserservice-Diskette erhältlich. In dieser Preislage (6,50 Mark und etwas Schweiß beim Eintippen, beziehungsweise 29,80 Mark für die Diskette) habe ich noch keinen besseren gesehen und auch fast alle kostspieligeren Assembler reichen ihm nicht das Wasser. Weil man aber auch einen Monitor braucht, findet zu diesem Zweck weiterhin der SMON hier seinen Platz. Ich verwende eine Version, die bei \$9000 beginnt, um ab \$C000 Raum zu lassen für den Reassembler zu Hypra-Ass (64'er, Ausgabe 11/85). Das ist ein nützliches Programm, mit dem man Maschinencode aus dem Speicher wieder in einen Quelltext umwandeln kann. Mittels des Hypra-Ass ist dieser Quelltext dann bearbeitbar. Mit diesem kompletten Instrumentarium sind wir allen Aufgaben gewachsen.

3. Einfache Schleifen

Eine der meistgebrauchten Strukturen in Basic und auch eine der wichtigsten in Assembler ist die Programmschleife. Als »einfache« Schleife bezeichne ich solche, die zum Zählen nur 1 Byte erfordern, also maximal 256 Durchläufe erlauben. Die verschiedenen Möglichkeiten sehen wir uns anhand von Verzögerungsschleifen an, die zunächst einmal nichts anderes tun, als zu zählen und Zeit zu verbrauchen (welch ein Luxus!). Die einfachste Variante lautet in Basic etwa:

```
10 FOR I = 0 TO 255
20 NEXT I
```

In Bild 1 finden Sie die »Übersetzung« in Assembler.

Sie können sowohl das Y-Register (Variante 1) als auch das X-Register (Variante 2) zum Zählen verwenden. In Zeile 5 finden Sie .LI 1,4. Das ist ein Pseudobefehl — also kein 6502-Befehl —, der die Ausgabe des Protokolls über den Drucker bewirkt. Zeile 40 enthält durch .BA \$5000 wieder einen Pseudobefehl. Damit legt man fest, von welcher Adresse an der Maschinencode in den Speicher gelegt werden soll. Die Zeilen 50 bis 90 sind unser

Assemblerprogramm. Zuerst wird ein Startwert 0 in das Y-Register geschrieben und dieses dann in der Zeile mit dem Label um 1 hochgezählt.

Falls Ihnen der Ausdruck Label noch nicht geläufig ist: Natürlich kann man auch statt dessen die Adresse 5002 hinter den BCC-Befehl in Zeile 80 schreiben — so haben wir das ja bisher immer mit dem SMON-Assembler getan. Das hätte aber im Quelltext, den wir hier schreiben, den Nachteil, daß wir diese Adresse jedesmal ändern müßten, wenn wir uns entschlossen, mit dem Pseudobefehl .BA den Programmstart zu verlegen. Indem wir aber diese Zeile durch das Label-Kennzeichen markieren, merkt sich der Hypra-Ass die dazugehörige Zeilennummer und rechnet sie beim Assemblieren automatisch in die richtige Sprungadresse um.

Ein weiterer Vorteil ist, daß man zu Dokumentationszwecken jede wichtige Adresse auf diese Weise markieren und sich am Schluß durch eine Symboltabelle ausgeben lassen kann. Besonders bei langen Programmen, in denen man dann sinnvolle Labelnamen verwendet (beispielsweise DRUCKEN am Anfang des Programmteils, das einen Ausdruck steuert), kann das eine unschätzbare Hilfe sein.

In unserem Programm in Bild 1 geht es weiter mit dem Vergleich, ob im Y-Register nach der Erhöhung schon \$FF erreicht wurde. Ist das nicht der Fall, dann ist das Carry-Bit frei und der Programmablauf verzweigt zurück zur Labelzeile. Ansonsten ist die Verzögerungsschleife beendet und mit dem BRK meldet sich der SMON, den Sie zu diesem Zeitpunkt natürlich im Speicher haben sollten (vergessen Sie nicht, den SMON zumindest einmal zu starten mit SYS »startadresse«, damit bei einem BRK in den SMON gesprungen wird).

Falls Sie das Programm durch SYS \$5000 vom Hypra-Ass aus gestartet haben (und nicht durch G 5000 aus dem SMON), finden Sie sich ebenfalls im Monitor wieder. Fast alle Beispielprogramme in diesem Kurs werden mit BRK enden. Der Grund dafür ist, daß es oft interessant ist, die Register nach dem Programmende zu beobachten. Sollten Sie ohne Monitor arbeiten wollen, dann müßten Sie statt dessen ein

RTS einsetzen. Hinter dem eigentlichen Programm finden Sie .SY 1,4. Auch das ist ein Pseudobefehl, der die Ausgabe der Symboltabelle über den Drucker bewirkt. Nicht sichtbar ist ein Befehl .ST, mit dem die Assemblierung beendet wird. Einige interessante Angaben besorgt uns der Hypra-Ass noch nach der kurzen Symboltabelle: Eine Zeitangabe und den Bereich, in dem der Maschinencode nun nach der Assemblierung zu finden ist. Falls Sie diesen Objektcode (so nennt man den Maschinencode auch häufig) speichern wollen (vom Monitor aus mit dem S-Kommando möglich), dann brauchen Sie diese Angaben. Unser Programm würde dann so abgespeichert:

S"OBJ.VERZ.VAR1" 5000 5008
(Man muß immer ein Byte zur Endadresse hinzurechnen beim Speichern des Objektcode). Eine andere Möglichkeit, den Objektcode auf Diskette zu speichern: Nach dem .LI 1,4 in der nächsten Zeile folgenden Befehl einsetzen:

10 -OB"OBJ.VERZ.VAR1,p,w
Jetzt wird nach dem Starten des Assemblierens mit RUN automatisch das Maschinenprogramm gespeichert.
Die weiteren Programmbeispiele werde ich nicht so erschöpfend erklären wie dieses. Nur wenn neue Pseudobefehle verwendet werden oder eine neue Programmstruktur das erfordert, geht's nochmal in die Tiefe. Um etwas Platz zu sparen, wurden die folgenden Programme nicht mit dem .LI - Befehl aus dem Drucker ausgegeben, sondern mit
OPEN 1,4:CMD1
/E

Dadurch werden die Adressen mit den Hex-Codes der Maschinenbefehle nicht gedruckt, sondern nur das Listing, wie es auch auf dem Bildschirm zu sehen ist.

Häufig tritt in Schleifen der Fall ein, daß weder das Y- noch das X-Register zur Verfügung stehen. Sie dienen dann anderweitig schon als Index. Statt dessen kann ebensogut eine Speicherstelle den Zähler bilden, wie in dieser Variante 3:

```
LDA # $00
STA $FB ;$FBist Zähler
LABEL INC $FB
LDA $FB
CMP # $FF
BCC LABEL
RTS
```

Selbstverständlich kann auch jede andere Speicherstelle anstelle von \$FB verwendet werden, sogar eine, die nicht in der Zeropage liegt. Voraussetzung ist lediglich, daß sie nicht innerhalb der Schleife verändert wird — außer zum Zählen der Schleifendurchläufe. In den bisher kennengelernten Varianten ha-

```
HYPR-ASS ASSEMBLERLISTING:
5 - .LI 1,4
;*** VERZOEGERUNGSSCHLEIFE VARIANTEN 1 UND 2 ***
; X- ODER Y-REGISTER ALS ZAEHLER
;
5000 A000 :50 - .BA $5000
5002 CB :60 -LDY # $00 ;BZW. LDY
5003 C0FF :70 -LDI # $00 ;INX
5005 90FB :80 -CPY # $FF ;CPX
5007 00 :90 -BCC LABEL ;WENN <255
:100 - .SY 1,4
BRK

SYMBOLS IN ALPHABETICAL ORDER:
LABEL = $5002
END OF ASSEMBLY 0:14.4
BASE = $5000 LAST BYTE AT $5007
```

Bild 1. Etwas zögern mit Variante 1 und 2

```
10 - .LI 1,4
20 - .BA $5000
30 -;*** VERZOEGERUNGSSCHLEIFE VARIANTE 4 ***
40 -;VARIABLER ENDWERT
50 -;Y-REGISTER ALS ZAEHLER
60 -;
70 - LDA # $20 ;DAS IST DEZIMAL 32
80 - STA $FA ;ENDWERT SPEICHERN
90 -;
100 - LDY # $00 ;Y-REGISTER INITIALISIEREN
110 -LDI # $00
120 -LDI # $00
130 -LDI # $00
140 -LDI # $00
150 -;
160 - .SY 1,4
170 - .ST
```

Bild 2. Verschieden zögern mit Variante 4

```
10 - .LI 1,4
20 - .BA $5000
30 -;*** VERZOEGERUNGSSCHLEIFE VARIANTE 5 ***
40 -;VARIABLER ENDWERT
50 -;SPEICHERSTELLE ALS ZAEHLER
60 -;
70 - LDA # $20 ;DAS IST DEZIMAL 32
80 - STA $FA ;ENDWERT SPEICHERN
90 -;
100 - LDA # $00 ;ZAEHLER INITIALISIEREN
110 - STA $5100 ;DA IST ER:UNSER ZAEHLER
120 -;
130 -LDI # $100
140 -LDI # $100
150 -LDI # $100
160 -LDI # $100
170 -LDI # $100
180 -LDI # $100
190 -LDI # $100
200 -LDI # $100
210 -LDI # $100
220 -LDI # $100
230 -LDI # $100
240 -LDI # $100
250 -LDI # $100
260 -LDI # $100
270 -LDI # $100
280 -LDI # $100
290 -LDI # $100
300 -LDI # $100
310 -LDI # $100
320 -LDI # $100
330 -LDI # $100
340 -LDI # $100
350 -LDI # $100
360 -LDI # $100
370 -LDI # $100
380 -LDI # $100
390 -LDI # $100
400 -LDI # $100
410 -LDI # $100
420 -LDI # $100
430 -LDI # $100
440 -LDI # $100
450 -LDI # $100
460 -LDI # $100
470 -LDI # $100
480 -LDI # $100
490 -LDI # $100
500 -LDI # $100
510 -LDI # $100
520 -LDI # $100
530 -LDI # $100
540 -LDI # $100
550 -LDI # $100
560 -LDI # $100
570 -LDI # $100
580 -LDI # $100
590 -LDI # $100
600 -LDI # $100
610 -LDI # $100
620 -LDI # $100
630 -LDI # $100
640 -LDI # $100
650 -LDI # $100
660 -LDI # $100
670 -LDI # $100
680 -LDI # $100
690 -LDI # $100
700 -LDI # $100
710 -LDI # $100
720 -LDI # $100
730 -LDI # $100
740 -LDI # $100
750 -LDI # $100
760 -LDI # $100
770 -LDI # $100
780 -LDI # $100
790 -LDI # $100
800 -LDI # $100
810 -LDI # $100
820 -LDI # $100
830 -LDI # $100
840 -LDI # $100
850 -LDI # $100
860 -LDI # $100
870 -LDI # $100
880 -LDI # $100
890 -LDI # $100
900 -LDI # $100
910 -LDI # $100
920 -LDI # $100
930 -LDI # $100
940 -LDI # $100
950 -LDI # $100
960 -LDI # $100
970 -LDI # $100
980 -LDI # $100
990 -LDI # $100
1000 -LDI # $100
1010 -LDI # $100
1020 -LDI # $100
1030 -LDI # $100
1040 -LDI # $100
1050 -LDI # $100
1060 -LDI # $100
1070 -LDI # $100
1080 -LDI # $100
1090 -LDI # $100
1100 -LDI # $100
1110 -LDI # $100
1120 -LDI # $100
1130 -LDI # $100
1140 -LDI # $100
1150 -LDI # $100
1160 -LDI # $100
1170 -LDI # $100
1180 -LDI # $100
1190 -LDI # $100
1200 -LDI # $100
1210 -LDI # $100
1220 -LDI # $100
1230 -LDI # $100
1240 -LDI # $100
1250 -LDI # $100
1260 -LDI # $100
1270 -LDI # $100
1280 -LDI # $100
1290 -LDI # $100
1300 -LDI # $100
1310 -LDI # $100
1320 -LDI # $100
1330 -LDI # $100
1340 -LDI # $100
1350 -LDI # $100
1360 -LDI # $100
1370 -LDI # $100
1380 -LDI # $100
1390 -LDI # $100
1400 -LDI # $100
1410 -LDI # $100
1420 -LDI # $100
1430 -LDI # $100
1440 -LDI # $100
1450 -LDI # $100
1460 -LDI # $100
1470 -LDI # $100
1480 -LDI # $100
1490 -LDI # $100
1500 -LDI # $100
1510 -LDI # $100
1520 -LDI # $100
1530 -LDI # $100
1540 -LDI # $100
1550 -LDI # $100
1560 -LDI # $100
1570 -LDI # $100
1580 -LDI # $100
1590 -LDI # $100
1600 -LDI # $100
1610 -LDI # $100
1620 -LDI # $100
1630 -LDI # $100
1640 -LDI # $100
1650 -LDI # $100
1660 -LDI # $100
1670 -LDI # $100
1680 -LDI # $100
1690 -LDI # $100
1700 -LDI # $100
1710 -LDI # $100
1720 -LDI # $100
1730 -LDI # $100
1740 -LDI # $100
1750 -LDI # $100
1760 -LDI # $100
1770 -LDI # $100
1780 -LDI # $100
1790 -LDI # $100
1800 -LDI # $100
1810 -LDI # $100
1820 -LDI # $100
1830 -LDI # $100
1840 -LDI # $100
1850 -LDI # $100
1860 -LDI # $100
1870 -LDI # $100
1880 -LDI # $100
1890 -LDI # $100
1900 -LDI # $100
1910 -LDI # $100
1920 -LDI # $100
1930 -LDI # $100
1940 -LDI # $100
1950 -LDI # $100
1960 -LDI # $100
1970 -LDI # $100
1980 -LDI # $100
1990 -LDI # $100
2000 -LDI # $100
2010 -LDI # $100
2020 -LDI # $100
2030 -LDI # $100
2040 -LDI # $100
2050 -LDI # $100
2060 -LDI # $100
2070 -LDI # $100
2080 -LDI # $100
2090 -LDI # $100
2100 -LDI # $100
2110 -LDI # $100
2120 -LDI # $100
2130 -LDI # $100
2140 -LDI # $100
2150 -LDI # $100
2160 -LDI # $100
2170 -LDI # $100
2180 -LDI # $100
2190 -LDI # $100
2200 -LDI # $100
2210 -LDI # $100
2220 -LDI # $100
2230 -LDI # $100
2240 -LDI # $100
2250 -LDI # $100
2260 -LDI # $100
2270 -LDI # $100
2280 -LDI # $100
2290 -LDI # $100
2300 -LDI # $100
2310 -LDI # $100
2320 -LDI # $100
2330 -LDI # $100
2340 -LDI # $100
2350 -LDI # $100
2360 -LDI # $100
2370 -LDI # $100
2380 -LDI # $100
2390 -LDI # $100
2400 -LDI # $100
2410 -LDI # $100
2420 -LDI # $100
2430 -LDI # $100
2440 -LDI # $100
2450 -LDI # $100
2460 -LDI # $100
2470 -LDI # $100
2480 -LDI # $100
2490 -LDI # $100
2500 -LDI # $100
2510 -LDI # $100
2520 -LDI # $100
2530 -LDI # $100
2540 -LDI # $100
2550 -LDI # $100
2560 -LDI # $100
2570 -LDI # $100
2580 -LDI # $100
2590 -LDI # $100
2600 -LDI # $100
2610 -LDI # $100
2620 -LDI # $100
2630 -LDI # $100
2640 -LDI # $100
2650 -LDI # $100
2660 -LDI # $100
2670 -LDI # $100
2680 -LDI # $100
2690 -LDI # $100
2700 -LDI # $100
2710 -LDI # $100
2720 -LDI # $100
2730 -LDI # $100
2740 -LDI # $100
2750 -LDI # $100
2760 -LDI # $100
2770 -LDI # $100
2780 -LDI # $100
2790 -LDI # $100
2800 -LDI # $100
2810 -LDI # $100
2820 -LDI # $100
2830 -LDI # $100
2840 -LDI # $100
2850 -LDI # $100
2860 -LDI # $100
2870 -LDI # $100
2880 -LDI # $100
2890 -LDI # $100
2900 -LDI # $100
2910 -LDI # $100
2920 -LDI # $100
2930 -LDI # $100
2940 -LDI # $100
2950 -LDI # $100
2960 -LDI # $100
2970 -LDI # $100
2980 -LDI # $100
2990 -LDI # $100
3000 -LDI # $100
3010 -LDI # $100
3020 -LDI # $100
3030 -LDI # $100
3040 -LDI # $100
3050 -LDI # $100
3060 -LDI # $100
3070 -LDI # $100
3080 -LDI # $100
3090 -LDI # $100
3100 -LDI # $100
3110 -LDI # $100
3120 -LDI # $100
3130 -LDI # $100
3140 -LDI # $100
3150 -LDI # $100
3160 -LDI # $100
3170 -LDI # $100
3180 -LDI # $100
3190 -LDI # $100
3200 -LDI # $100
3210 -LDI # $100
3220 -LDI # $100
3230 -LDI # $100
3240 -LDI # $100
3250 -LDI # $100
3260 -LDI # $100
3270 -LDI # $100
3280 -LDI # $100
3290 -LDI # $100
3300 -LDI # $100
3310 -LDI # $100
3320 -LDI # $100
3330 -LDI # $100
3340 -LDI # $100
3350 -LDI # $100
3360 -LDI # $100
3370 -LDI # $100
3380 -LDI # $100
3390 -LDI # $100
3400 -LDI # $100
3410 -LDI # $100
3420 -LDI # $100
3430 -LDI # $100
3440 -LDI # $100
3450 -LDI # $100
3460 -LDI # $100
3470 -LDI # $100
3480 -LDI # $100
3490 -LDI # $100
3500 -LDI # $100
3510 -LDI # $100
3520 -LDI # $100
3530 -LDI # $100
3540 -LDI # $100
3550 -LDI # $100
3560 -LDI # $100
3570 -LDI # $100
3580 -LDI # $100
3590 -LDI # $100
3600 -LDI # $100
3610 -LDI # $100
3620 -LDI # $100
3630 -LDI # $100
3640 -LDI # $100
3650 -LDI # $100
3660 -LDI # $100
3670 -LDI # $100
3680 -LDI # $100
3690 -LDI # $100
3700 -LDI # $100
3710 -LDI # $100
3720 -LDI # $100
3730 -LDI # $100
3740 -LDI # $100
3750 -LDI # $100
3760 -LDI # $100
3770 -LDI # $100
3780 -LDI # $100
3790 -LDI # $100
3800 -LDI # $100
3810 -LDI # $100
3820 -LDI # $100
3830 -LDI # $100
3840 -LDI # $100
3850 -LDI # $100
3860 -LDI # $100
3870 -LDI # $100
3880 -LDI # $100
3890 -LDI # $100
3900 -LDI # $100
3910 -LDI # $100
3920 -LDI # $100
3930 -LDI # $100
3940 -LDI # $100
3950 -LDI # $100
3960 -LDI # $100
3970 -LDI # $100
3980 -LDI # $100
3990 -LDI # $100
4000 -LDI # $100
4010 -LDI # $100
4020 -LDI # $100
4030 -LDI # $100
4040 -LDI # $100
4050 -LDI # $100
4060 -LDI # $100
4070 -LDI # $100
4080 -LDI # $100
4090 -LDI # $100
4100 -LDI # $100
4110 -LDI # $100
4120 -LDI # $100
4130 -LDI # $100
4140 -LDI # $100
4150 -LDI # $100
4160 -LDI # $100
4170 -LDI # $100
4180 -LDI # $100
4190 -LDI # $100
4200 -LDI # $100
4210 -LDI # $100
4220 -LDI # $100
4230 -LDI # $100
4240 -LDI # $100
4250 -LDI # $100
4260 -LDI # $100
4270 -LDI # $100
4280 -LDI # $100
4290 -LDI # $100
4300 -LDI # $100
4310 -LDI # $100
4320 -LDI # $100
4330 -LDI # $100
4340 -LDI # $100
4350 -LDI # $100
4360 -LDI # $100
4370 -LDI # $100
4380 -LDI # $100
4390 -LDI # $100
4400 -LDI # $100
4410 -LDI # $100
4420 -LDI # $100
4430 -LDI # $100
4440 -LDI # $100
4450 -LDI # $100
4460 -LDI # $100
4470 -LDI # $100
4480 -LDI # $100
4490 -LDI # $100
4500 -LDI # $100
4510 -LDI # $100
4520 -LDI # $100
4530 -LDI # $100
4540 -LDI # $100
4550 -LDI # $100
4560 -LDI # $100
4570 -LDI # $100
4580 -LDI # $100
4590 -LDI # $100
4600 -LDI # $100
4610 -LDI # $100
4620 -LDI # $100
4630 -LDI # $100
4640 -LDI # $100
4650 -LDI # $100
4660 -LDI # $100
4670 -LDI # $100
4680 -LDI # $100
4690 -LDI # $100
4700 -LDI # $100
4710 -LDI # $100
4720 -LDI # $100
4730 -LDI # $100
4740 -LDI # $100
4750 -LDI # $100
4760 -LDI # $100
4770 -LDI # $100
4780 -LDI # $100
4790 -LDI # $100
4800 -LDI # $100
4810 -LDI # $100
4820 -LDI # $100
4830 -LDI # $100
4840 -LDI # $100
4850 -LDI # $100
4860 -LDI # $100
4870 -LDI # $100
4880 -LDI # $100
4890 -LDI # $100
4900 -LDI # $100
4910 -LDI # $100
4920 -LDI # $100
4930 -LDI # $100
4940 -LDI # $100
4950 -LDI # $100
4960 -LDI # $100
4970 -LDI # $100
4980 -LDI # $100
4990 -LDI # $100
5000 -LDI # $100
5010 -LDI # $100
5020 -LDI # $100
5030 -LDI # $100
5040 -LDI # $100
5050 -LDI # $100
5060 -LDI # $100
5070 -LDI # $100
5080 -LDI # $100
5090 -LDI # $100
5100 -LDI # $100
5110 -LDI # $100
5120 -LDI # $100
5130 -LDI # $100
5140 -LDI # $100
5150 -LDI # $100
5160 -LDI # $100
5170 -LDI # $100
5180 -LDI # $100
5190 -LDI # $100
5200 -LDI # $100
5210 -LDI # $100
5220 -LDI # $100
5230 -LDI # $100
5240 -LDI # $100
5250 -LDI # $100
5260 -LDI # $100
5270 -LDI # $100
5280 -LDI # $100
5290 -LDI # $100
5300 -LDI # $100
5310 -LDI # $100
5320 -LDI # $100
5330 -LDI # $100
5340 -LDI # $100
5350 -LDI # $100
5360 -LDI # $100
5370 -LDI # $100
5380 -LDI # $100
5390 -LDI # $100
5400 -LDI # $100
5410 -LDI # $100
5420 -LDI # $100
5430 -LDI # $100
5440 -LDI # $100
5450 -LDI # $100
5460 -LDI # $100
5470 -LDI # $100
5480 -LDI # $100
5490 -LDI # $100
5500 -LDI # $100
5510 -LDI # $100
5520 -LDI # $100
5530 -LDI # $100
5540 -LDI # $100
5550 -LDI # $100
5560 -LDI # $100
5570 -LDI # $100
5580 -LDI # $100
5590 -LDI # $100
5600 -LDI # $100
5610 -LDI # $100
5620 -LDI # $100
5630 -LDI # $100
5640 -LDI # $100
5650 -LDI # $100
5660 -LDI # $100
5670 -LDI # $100
5680 -LDI # $100
5690 -LDI # $100
5700 -LDI # $100
5710 -LDI # $100
5720 -LDI # $100
5730 -LDI # $100
5740 -LDI # $100
5750 -LDI # $100
5760 -LDI # $100
5770 -LDI # $100
5780 -LDI # $100
5790 -LDI # $100
5800 -LDI # $100
5810 -LDI # $100
5820 -LDI # $100
5830 -LDI # $100
5840 -LDI # $100
5850 -LDI # $100
5860 -LDI # $100
5870 -LDI # $100
5880 -LDI # $100
5890 -LDI # $100
5900 -LDI # $100
5910 -LDI # $100
5920 -LDI # $100
5930 -LDI # $100
5940 -LDI # $100
5950 -LDI # $100
5960 -LDI # $100
5970 -LDI # $100
5980 -LDI # $100
5990 -LDI # $100
6000 -LDI # $100
6010 -LDI # $100
6020 -LDI # $100
6030 -LDI # $100
6040 -LDI # $100
6050 -LDI # $100
6060 -LDI # $100
6070 -LDI # $100
6080 -LDI # $100
6090 -LDI # $100
6100 -LDI # $100
6110 -LDI # $100
6120 -LDI # $100
6130 -LDI # $100
6140 -LDI # $100
6150 -LDI # $100
6160 -LDI # $100
6170 -LDI # $100
6180 -LDI # $100
6190 -LDI # $100
6200 -LDI # $100
6210 -LDI # $100
6220 -LDI # $100
6230 -LDI # $100
6240 -LDI # $100
6250 -LDI # $100
6260 -LDI # $100
6270 -LDI # $100
6280 -LDI # $100
6290 -LDI # $100
6300 -LDI # $100
6310 -LDI # $100
6320 -LDI # $100
6330 -LDI # $100
6340 -LDI # $100
6350 -LDI # $100
6360 -LDI # $100
6370 -LDI # $100
6380 -LDI # $100
6390 -LDI # $100
6400 -LDI # $100
6410 -LDI # $100
6420 -LDI # $100
6430 -LDI # $100
6440 -LDI # $100
6450 -LDI # $100
6460 -LDI # $100
6470 -LDI # $100
6480 -LDI # $100
6490 -LDI # $100
6500 -LDI # $100
6510 -LDI # $100
6520 -LDI # $100
6530 -LDI # $100
6540 -LDI # $100
6550 -LDI # $100
6560 -LDI # $100
6570 -LDI # $100
6580 -LDI # $100
6590 -LDI # $100
6600 -LDI # $100
6610 -LDI # $100
6620 -LDI # $100
6630 -LDI # $100
6640 -LDI # $100
6650 -LDI # $100
6660 -LDI # $100
6670 -LDI # $100
6680 -LDI # $100
6690 -LDI # $100
6700 -LDI # $100
6710 -LDI # $100
6720 -LDI # $100
6730 -LDI # $100
6740 -LDI # $100
6750 -LDI # $100
6760 -LDI # $100
6770 -LDI # $100
6780 -LDI # $100
6790 -LDI # $100
6800 -LDI # $100
6810 -LDI # $100
6820 -LDI # $100
6830 -LDI # $100
6840 -LDI # $100
6850 -LDI # $100
6860 -LDI # $100
6870 -LDI # $100
6880 -LDI # $100
6890 -LDI # $100
6900 -LDI # $100
6910 -LDI # $100
6920 -LDI # $100
6930 -LDI # $100
6940 -LDI # $100
6950 -LDI # $100
6960 -LDI # $100
6970 -LDI # $100
6980 -LDI # $100
6990 -LDI # $100
7000 -LDI # $100
7010 -LDI # $100
7020 -LDI # $100
7030 -LDI # $100
7040 -LDI # $100
7050 -LDI # $100
7060 -LDI # $100
7070 -LDI # $100
7080 -LDI # $100
7090 -LDI # $100
7100 -LDI # $100
7110 -LDI # $100
7120 -LDI # $100
7130 -LDI # $100
7140 -LDI # $100
7150 -LDI # $100
7160 -LDI # $100
7170 -LDI # $100
7180 -LDI # $100
7190 -LDI # $100
7200 -LDI # $100
7210 -LDI # $100
7220 -LDI # $100
7230 -LDI # $100
7240 -LDI # $100
7250 -LDI # $100
7260 -LDI # $100
7270 -LDI # $100
7280 -LDI # $100
7290 -LDI # $100
7300 -LDI # $100
7310 -LDI # $100
7320 -LDI # $100
7330 -LDI # $100
7340 -LDI # $100
7350 -LDI # $100
7360 -LDI # $100
7370 -LDI # $100
7380 -LDI # $100
7390 -LDI # $100
7400 -LDI # $100
7410 -LDI # $100
7420 -LDI # $100
7430 -LDI # $100
7440 -LDI # $100
7450 -LDI # $100
7460 -LDI # $100
7470 -LDI # $100
7480 -LDI # $100
7490 -LDI # $100
7500 -LDI # $100
7510 -LDI # $100
7520 -LDI # $100
7530 -LDI # $100
7540 -LDI # $100
7550 -LDI # $100
7560 -LDI # $100
7570 -LDI # $100
7580 -LDI # $100
7590 -LDI # $100
7600 -LDI # $100
7610 -LDI # $100
7620 -LDI # $100
7630 -LDI # $100
7640 -LDI # $100
7650 -LDI # $100
7660 -LDI # $100
7670 -LDI # $100
7680 -LDI # $100
7690 -LDI # $100
7700 -LDI # $100
7710 -LDI # $100
7720 -LDI # $100
7730 -LDI # $100
7740 -LDI # $100
7750 -LDI # $100
7760 -LDI # $100
7770 -LDI # $100
7780 -LDI # $100
7790 -LDI # $100
7800 -LDI # $100
7810 -LDI # $100
7820 -LDI # $100
7830 -LDI # $100
7840 -LDI # $100
7850 -LDI # $100
7860 -LDI # $100
7870 -LDI # $100
7880 -LDI # $100
7890 -LDI # $100
7900 -LDI # $100
7910 -LDI # $100
7920 -LDI # $100
7930 -LDI # $100
7940 -LDI # $100
7950 -LDI # $100
7960 -LDI # $100
7970 -LDI # $100
7980 -LDI # $100
7990 -LDI # $100
8000 -LDI # $100
8010 -LDI # $100
8020 -LDI # $100
8030 -LDI # $100
8040 -LDI # $100
8050 -LDI # $100
8060 -LDI # $100
8070 -LDI # $100
8080 -LDI # $100
8090 -LDI # $100
8100 -LDI # $100
8110 -LDI # $100
8120 -LDI # $100
8130 -LDI # $100
8140 -LDI # $100
8150 -LDI # $100
8160 -LDI # $100
8170 -LDI # $100
8180 -LDI # $100
8190 -LDI # $100
8200 -LDI # $100
8210 -LDI # $100
8220 -LDI # $100
8230 -LDI # $100
8240 -LDI # $100
8250 -LDI # $100
8260 -LDI # $100
8270 -LDI # $100
8280 -LDI # $100
8290 -LDI # $100
8300 -LDI # $100
8310 -LDI # $100
8320 -LDI # $100
8330 -LDI # $100
8340 -LDI # $100
8350 -LDI # $100
8360 -LDI # $100
8370 -LDI # $100
8380 -LDI # $100
8390 -LDI # $100
8400 -LDI # $100
8410 -LDI # $100
8420 -LDI # $100
8430 -LDI # $100
8440 -
```

In Bild 5 finden Sie diese Version. \$FA dient als Zähler.

Etwas schwieriger wird die Programmierung, wenn man nicht nur um 1 herauf- oder herunterzählt, sondern um 2,3,4 oder mehr. Das Basic-Äquivalent drückt sich dann beispielsweise in der Ergänzung STEP -2 der FOR...NEXT-Schleife aus. Dreht es sich nur um kleine Schrittweiten, die konstant bleiben, dann verwendet man vorzuziehend mehrere DEY (oder DEX, DEC, INY, INX und INC) hintereinander. Man muß außerdem mit der Abbruchbedingung einer solchen Schleife vorsichtig sein. BNE ist nicht immer möglich, weil man unter Umständen schon vor der Prüfung (durch BNE) unter 0 hindurchgezählt hat (dann folgt ja wieder \$FF etc.). Hat beispielsweise der Zähler (hier das Y-Register) den Wert 1 und es werde durch eine Sequenz:

```
DEY
DEY
DEY
BNE LABEL
```

weitergezählt, dann nimmt Y der Reihe nach die Werte 0,FF,FE an und BNE findet die Zeroflagge nicht gesetzt. Man muß also andere Abbruchbedingungen verwenden. Solange man bis zur ersten Prüfung (also dem ersten Schleifendurchlauf beim Herunterzählen) im Zähler mindestens \$7F (=binär 0111 1111) vorliegen hat, kann man mittels BPL die Schleife schließen. Zur Erinnerung: BPL verzweigt, wenn Bit 7 nicht gesetzt ist (kleiner 128), BMI verzweigt, wenn Bit 7 gesetzt ist (größer oder gleich 128). Das Basic-Programmstück

```
10 FOR I = 32 TO 0 STEP -3
20 NEXT I
```

findet seine Entsprechung in dem Assemblerlisting Version 8 in Bild 6.

Wieder dient das Y-Register als Schleifenzähler.

Größere Schrittweiten lassen es – von einer gewissen Grenze an, die durch das Verhältnis von Bytezahl auf der einen und Bearbeitungsdauer auf der anderen Seite, bestimmt wird – sinnvoll erscheinen, den Zähler durch Subtraktion (oder Addition beim Aufwärtszählen) zu verändern. Das Analogon zur Basic-Sequenz:

```
10 FOR I = 127 TO 0 STEP -10
20 NEXT I
```

sehen Sie in Bild 7.

In dieser Version 9 dient die Zeropagespeicherstelle \$FA als Zähler und in den Programmzeilen 80 bis 110 findet die Verminderung dieses Zählers durch Subtraktion statt (\$0A = dezimal 10). Das Programm kann noch verändert werden, indem man anstelle von BPL den BCS-Befehl verwendet. Wenn die Subtraktion einen Unterlauf ergeben hat, wird das Carry-Bit gelöscht.

```
10 - .LI 1,4
20 - .BA $5000
30 -;*** VERZOEGERUNGSSCHLEIFE VERSION 8 ***
40 -;FOR I = 32 TO 0 STEP-3
50 -;
60 - LDY #$20 ;STARTWERT IN ZAEHLER
70 -LABEL DEY ;MINUS 3
80 - DEY
90 - DEY
100 - BPL LABEL ;WEITER BIS UNTERLAUF
110 - BRK
120 -;
130 - .SY 1,4
140 - .ST
```

Bild 6. Verzögern in kleinen Schritten mittels Variante 8

```
10 - .LI 1,4
20 - .BA $5000
30 -;*** VERZOEGERUNGSSCHLEIFE VERSION 9 ***
40 -;FOR I = 127 TO 0 STEP -10
50 -;
60 - LDA #$7F ;DAS IST DEZIMAL 127
70 - STA $FA ; UNSER ZAEHLER
80 -LABEL SEC
90 - LDA $FA
100 - SBC #$0A
110 - STA $FA
120 - BPL LABEL ;WEITER BIS UNTERLAUF
130 - BRK
140 - .SY 1,4
150 - .ST
```

Bild 7. Verzögern in großen Schritten mit Variante 9

Außerdem lassen sich noch 2 Byte einsparen, indem man das STA \$FA aus Zeile 110 herausnimmt und dafür das LABEL eine Zeile höher setzt. Allerdings geht das dann auf Kosten der Durchschaubarkeit unseres Programmes.

Wir wollen nun mit den einfachen Verzögerungsschleifen aufhören. Es gäbe noch weitere Aufgaben zu lösen (nämlich beispielsweise von einem bestimmten Startwert bis zu einem bestimmten Zielwert zu zählen), die vertraue ich aber Ihnen selbst an: Alles notwendige dazu können Sie aus unseren verschiedenen Versionen entnehmen und kombinieren. Interessant werden Schleifen hauptsächlich durch einen Job, der in ihnen

wiederholt ausgeführt wird. Zwei Beispiele sollen uns zur Illustration in dieser Folge dienen. Vorher aber sollen noch einige Bemerkungen zur grundsätzlichen Architektur von Schleifen gemacht werden.

Im Prinzip setzt sich jede Schleife aus vier Bestandteilen zusammen.

Initialisierung. Beispielsweise wird hier der Startwert des Zählers festgelegt.

Verarbeitung. Das ist das, was in den Verzögerungsschleifen bisher leer blieb: Der Job.

Steuerung. Hoch- oder Herunterzählen des Zählers und Prüfen der Abbruchbedingung.

Ausgang. Das war bisher bei uns immer der BRK-Befehl.

Aus diesen vier Bestandteilen

lassen sich zwei grundsätzliche Schleifenmöglichkeiten konstruieren, die Sie in Bild 8 dargestellt finden.

In Bild 8a haben wir das Prinzip vorliegen, das unseren normalen FOR...NEXT-Schleifen in Basic zugrundeliegt. Diese Schleife wird mindestens einmal durchlaufen. Erst nach der Ausführung des Jobs erfolgt die Prüfung, ob die Abbruchbedingung gegeben war. Soll solch eine Schleife n-mal durchlaufen werden, muß die Initialisierung mit n-1 im Zähler erfolgen (oder die Abbruchbedingung entsprechend umgeformt werden).

Die Schleifenkonstruktion in Bild 8b dagegen muß nicht durchlaufen werden. Ihr entspricht etwa eine DO UNTIL...LOOP-Schleife oder auch eine DO WHILE...LOOP-Schleife aus dem Basic 7.0 des C 128. Hier erfolgt die Initialisierung des Zählers genau mit dem Wert n.

Sehen wir uns beispielsweise unsere Version 9 an, dann entdecken wir die einzelnen Schleifenteile wie folgt:

Initialisierung:	LDA #\$7F STA \$FA
Verarbeitung:	LABEL
Steuerung:	SEC LDA \$FA SBC #\$0A STA \$FA BCS LABEL
Ausgang:	BRK

Auf diese Weise ist es Ihnen möglich, alle bisher kennengelernten Schleifenvarianten mit einem beliebigen Job zu füllen. Noch eines gibt es zu bedenken: Alle Instruktionen zwischen dem Label und der Abbruchbedingung werden oft ausgeführt, sind also zeitraubend. Daher sollte der auszuführende Job alle Befehle vermeiden, die ebensogut vor der eigentlichen Schleife stehen könnten.

Sehen wir uns unser 1. Beispiel an. Wir stellen uns die Aufgabe, von den 127 Zeichen, die mittels POKE-Code erfaßbar sind, jedes 2. Zeichen an jeder 2. Bildschirmstelle abzubilden. Das Ganze soll durch Verwenden verschiedener Farben auch noch hübsch bunt aussehen. In Basic würden wir dafür schreiben:

```
10 S = 1024 : C = 55296
20 FOR I = 127 TO 0 STEP -2
30 POKE S+I,I
40 POKE C+I,I
50 NEXT I
```

Weil im Bildschirmfarbspeicher nur die Bits 0 bis 3 eine Rolle spielen (die anderen aber gar nicht beachtet werden), erzeugen wir in Zeile 40 auch die verschiedenen Farben mehrmals nacheinander.

(Heimo Ponnath/gk)
Fortsetzung folgt.

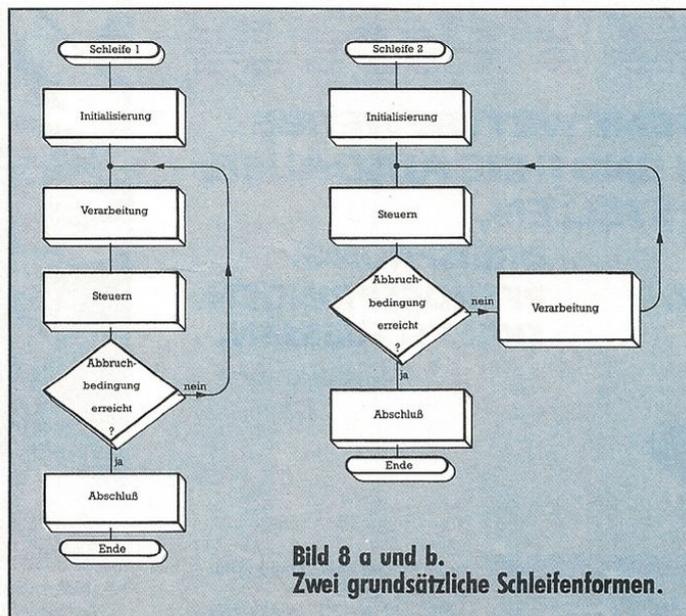


Bild 8 a und b. Zwei grundsätzliche Schleifenformen.