

```

,5000 A2 00 LDX #00
,5002 86 5C STX 5C
,5004 86 5D STX 5D
,5006 A0 10 LDY #10
,5008 06 57 ASL 57
,500A 26 58 ROL 58
,500C 26 5C ROL 5C
,500E 26 5D ROL 5D
,5010 38 SEC
,5011 A5 5C LDA 5C
,5013 E5 59 SBC 59
,5015 AA TAX
,5016 A5 5D LDA 5D
,5018 E5 5A SBC 5A
,501A 90 06 BCC 5022
,501C 86 5C STX 5C
,501E 85 5D STA 5D
,5020 E6 57 INC 57
,5022 88 DEY
,5023 D0 E3 BNE 5008
,5025 60 RTS
    
```

Programm 1. Die 16-Bit-Division

```

,6000 A9 00 LDA #00
,6002 85 5F STA 5F
,6004 A9 04 LDA #04
,6006 85 60 STA 60
,6008 A9 E8 LDA #E8
,600A 85 5A STA 5A
,600C 85 58 STA 58
,600E A9 07 LDA #07
,6010 85 5B STA 5B
,6012 A9 E3 LDA #E3
,6014 85 59 STA 59
,6016 20 BF A3 JSR A3BF
,6019 A9 00 LDA #00
,601B 85 5F STA 5F
,601D A9 D8 LDA #D8
,601F 85 60 STA 60
,6021 A9 E8 LDA #E8
,6023 85 5A STA 5A
,6025 A9 DB LDA #DB
,6027 85 5B STA 5B
,6029 A9 D1 LDA #D1
,602B 85 58 STA 58
,602D A9 E7 LDA #E7
,602F 85 59 STA 59
,6031 20 BF A3 JSR A3BF
,6034 A9 40 LDA #40
,6036 8D 11 03 STA 0311
,6039 A9 60 LDA #60
,603B 8D 12 03 STA 0312
,603E 60 RTS
    
```

Programm 3. Zweiter Teil der Verschieberoutine

```

$A004 CBMBASIC
$A09E - $A19D Texte der Basic-Befehls-
                worte
                (im letzten Byte ist jeweils Bit 7 gesetzt)
$A19E - $A327 Texte der Basic-Fehler- und System-
                Meldungen. (Im letzten Byte ist jeweils Bit 7 ge-
                setzt)
$A364 - $A38A Weitere System-Meldungen: OK, ERROR, IN,
                READY, BREAK. (Das letzte Byte ist jeweils 0)
$ACFC - $AD1D Fehlermeldungstexte für INPUT: ?EXTRA
                IGNORED, ?REDO FROM START. (Das letzte
                Byte ist jeweils 0)
$E460 BASIC BYTES FREE
$E473 **** COMMODORE 64 BASIC V2 ****
                64K-RAM-System
$ECE6 LOAD (Return) RUN (Return)
$F0BD - $F12B Texte für Ein- und Ausgabe-Operationen
$FD10 CBM80
    
```

Tabelle 2. Diese Texte sind im ROM als ASCII-Werte abgelegt

Befehls- wort	Adressierung	Byte- zahl	Code		Takt- zyklen	Beein- flussg. von Flag- gen
			Hex	Dez		
LSR	»Akkumulator«	1	1A	26	2	N,ZC
	absolut	3	4E	78	6	N,ZC
	0-page-absolut	2	46	70	5	N,ZC
	absolut-X-indiz.	3	5E	94	7	N,ZC
ROL	0-page-X-indiz.	2	56	86	6	N,ZC
	»Akkumulator«	1	2A	42	2	N,ZC
	absolut	3	2E	46	6	N,ZC
	0-page-absolut	2	26	38	5	N,ZC
ROR	absolut-X-indiz.	3	3E	62	7	N,ZC
	0-page-X-indiz.	2	36	54	6	N,ZC
	»Akkumulator«	1	6A	106	2	N,ZC
	absolut	3	6E	110	6	N,ZC
ROR	0-page-absolut	2	66	102	5	N,ZC
	absolut-X-indiz.	3	7E	126	7	N,ZC
	0-page-X-indiz.	2	76	118	6	N,ZC

Tabelle 3. Die in dieser Ausgabe besprochenen Assembler-Befehle

Sollten Sie mal in die Verlegenheit kommen, solche Texte ausgeben zu wollen, dann legen Sie sie nicht nochmal in einer eigenen Texttafel ab, sondern schöpfen Sie aus dem Fundus, den wir im ROM-Bereich fix und fertig haben.

Diese Folge soll nicht abgeschlossen werden, ohne eine Korrektur. Auf einen Fehler, dem ich aufgesessen bin (in der Literatur befinde ich mich aber in guter Gesellschaft, andere sind auch davon betroffen), haben mich zwei aufmerksame Leser hingewiesen. Es dreht sich um die Flaggensetzung bei Compare-Befehlen. Die N-Flagge ist nämlich nicht nur vom Ergebnis des Vergleichs, sondern auch noch von den aktuellen Akku- beziehungsweise Registerinhalten bestimmt.

Bild 1 in der 5. Folge muß deshalb korrigiert werden: (A,X,Y) größer als die Daten: N kann 0 oder 1 sein (A,X,Y) = Daten N = 0 (A,X,Y) kleiner als die Daten: N kann 0 oder 1 sein.

Das stammt aus dem offiziellen MOS-Technology-Handbuch und entspricht somit hoffentlich der Wahrheit [2]. Das bedeutet, daß man bei den Abfragen durch Branch-Befehle nach den Vergleichsbefehlen etwas vorsichtig sein sollte, was die N-Flagge angeht.

Zum Schluß noch, wie üblich, die Tabelle 3 mit den neuen Assembler-Befehlen.

(Heimo Ponnath/gk)

[1] »Computerspiele und Wissenswertes Commodore 64«, Haar bei München: Markt & Technik Verlag, 1984. Das ist die von P. Lücke besorgte Übersetzung des amerikanischen Buches »More on the sixtyfour« und ist jedem Assembler-Programmierer warm zu empfehlen.

[2] »MOS Microcomputers Programmier-Handbuch«, Frankfurt: Commodore MOS Technology

Tips & Tricks gesucht



Jeder Computer und jedes Programm hat seine speziellen Schwachstellen und Unzulänglichkeiten. Allerdings ist kaum ein Programmierer oder Anwender auf Dauer bereit, sich damit abzufinden. Wo auch sorgfältigste Lektüre von Handbüchern nicht weiterhilft, da wird so manche Stunde experimentiert, um eine Lösung zu finden (die oft in einer Basic-Zeile Platz hat).

Wir suchen solche Tips und Tricks, um sie

allen Lesern zugänglich zu machen. Schließlich ist es wenig sinnvoll, sich wochenlang mit Problemen herumzuschlagen, die andere bereits gelöst haben.

Wenn Sie also interessante Tips für den Umgang mit Computer, Floppy, Drucker oder sonstiger Hardware haben, wenn Sie bei kommerzieller Software einige Kniffe kennen, die nicht in der Anleitung stehen, oder wenn Sie interessante Problemlö-

sungen statt in vier Seiten Listing in ein oder zwei Basic-Zeilen untergebracht haben, dann sollten Sie uns auf jeden Fall einmal schreiben.

Bitte geben Sie genau den Computertyp und die Gerätekonfiguration oder die Software an, und senden Sie Ihren Tip oder Trick an die

Redaktion 64'er
Markt & Technik Verlag
Aktiengesellschaft
Hans-Pinsel-Str. 2
8013 Haar bei München