

Ein 6502-Simulator in Basic

Der Simulator ist eine Hilfe für den Einstieg in die Maschinensprache-Programmierung auf dem VC20/C16. Man kann direkt am Bildschirm verfolgen, was sich bei der Abarbeitung eines Programms in der Zentraleinheit des Computers, im 6502/7501-Prozessor tut.

Die meisten Leser, die sich für Maschinensprache interessieren, werden mit Basic vertraut sein, aber in der Regel noch ein paar Schwierigkeiten mit der Funktion einzelner Maschinenbefehle haben. Wir wollen deshalb im folgenden ein Basic-Programm vorstellen, welches die Maschinenbefehle simuliert. Den Prozessorregistern entsprechen dabei einfache Variablen, und der Stapel wird mit Hilfe eines Arrays simuliert. Die momentanen Werte der Prozessorregister werden am Bildschirm in hexadezimaler Form ausgegeben. Durch die Verwendung von Basic dauert die Ausführung eines Befehls etwa 100000mal länger, als der Prozessor selbst arbeitet. Dies ist aber gerade dazu gut, um in einer Art Einzelschrittverfahren die Funktion eines Programms nachzuvollziehen.

Direkt nach dem Starten fragt das Programm nach der Startadresse, ab der ein Programm bearbeitet werden soll. Das zu testende Maschinenprogramm muß also bereits im Speicher stehen. Man kann sowohl eingebaute ROM-Routinen testen, als auch mit einem Assembler erstellte Programme nachvollziehen.

Auch ist es möglich, kleine Programme mit POKE-Befehlen einzugeben und dann ab der entsprechenden Speicherzelle den Test zu starten.

Im Anschluß werden die Werte der Prozessorregister, die an das Programm übergeben werden sollen, erfragt. Dabei kann man den Wert des Prozessorstatusregisters in binärer Form eingeben. Schließlich werden noch bis zu 10 Adressen erfragt, bei denen das Programm anhalten soll. Wird im Programmablauf eine dieser Adressen erreicht, so hält das Programm an. Alle Eingaben – außer den Daten des Prozessorstatusregisters – müssen in hexadezimaler Form erfolgen.

So funktioniert der 6502-Simulator

Nun wird das Programm schrittweise abgearbeitet und der aktuelle Stand der Prozessorregister nebeneinander auf dem Bildschirm angezeigt. Durch Drücken der SHIFT-Taste kann man das Programm kurzzeitig anhalten.

Das Programm ist wie folgt aufgebaut:

Die Zeilen 10 bis 90 geben die Kopfzeile des Programms aus und beinhalten die Eingaben. Dort werden auch einige Felder und Funktionen definiert, die zu späteren Berechnungen notwendig sind. In Zeile 70 befindet sich der Aufruf eines Unterprogramms ab Zeile 49000, welches ein Demonstrationsprogramm in die Speicherzellen ab 34768=\$8700 »POKEd«. In den Zeilen 100 bis 900 sind einige Unterprogramme enthalten, die der Umrechnung von Hexadezimal-

zahlen dienen, die hier nicht weiter erläutert werden. Ab Zeile 1000 bis 1060 befindet sich die Schleife, die für jeden Befehl durchlaufen wird. Dabei wird zunächst der aktuelle Stand der Prozessorregister angezeigt (Unterprogrammaufruf 2000), dann die Variable OP mit dem Wert der momentan über den Programmzähler adressierten Speicherzelle besetzt und anschließend das Unterprogramm ab Zeile 1100 aufgerufen, welches einen Befehl auswertet. Schließlich wird noch der Programmzähler um eins erhöht und wieder zur Zeile 1000 gesprungen, wenn kein Abbruchkriterium erfüllt ist.

Das Unterprogramm ab Zeile 1100 ist ein Sprungverteiler auf die einzelnen Programmstücke, jeweils ein Sprung für einen Befehl. Ein unzulässiger Befehl bedeutet immer einen Sprung auf das Unterprogramm ab Zeile 1400, welches mit einer entsprechenden Meldung am Bildschirm endet.

Das Unterprogramm ab Zeile 2000 dient zur Anzeige aller Prozessorregister. Das Statusregister wird dabei binär dargestellt, wobei ein Zeichen durch die Funktion CHR\$(48+N) gebildet wird, was hier einer »0« oder »1« entspricht, je nachdem, ob N gleich 0 oder 1 ist. Die anderen Prozessorregister werden mit Hilfe des Unterprogramms ab Zeile 200 in hexadezimale Zahlen umgewandelt. Diese werden gesammelt in Zeile 2090 in der zweiten Bildschirmzeile ausgegeben.

Die Zeilen 3000 bis 9000 enthalten Unterprogramme, die für die Ausführung von Befehlen notwendig sind, beziehungsweise allgemeine Unterprogramme, die die Programmierung der anderen vereinfachen.

Ab Zeile 3000 wird ein 1-Byte-Operand geholt, was dadurch geschieht, daß der Programmzähler um 1 erhöht wird und der Variablen O der Wert der angesprochenen Speicherzelle zugewiesen wird. O soll hier die Abkürzung für 1 Byte-Operand sein. Im Unterprogramm ab 4000 wird entsprechend ein 2-Byte-Operand geholt, der in der Variablen OO abgelegt wird.

Die Unterprogramme ab den Zeilen 5000 bis 6000 fassen die einzelnen Statusflags zur Variablen P zusammen beziehungsweise zerlegen diese.

Ab Zeile 7000 steht das Unterprogramm für die Ausführung eines relativen Sprunges.

Komplizierte Addition

Da die Addition ein komplizierter Prozeß ist, da insbesondere Binär- oder Dezimalarithmetik verwendet werden kann, wird diese in dem Unterprogramm ab Zeile 8000 behandelt. Analog wird ab 8200 die Subtraktion behandelt. In diesen Unterprogrammen wird zunächst erfragt, ob das Dezimalflag gesetzt ist. Wenn nicht, ergibt sich der neue Wert des Akkumulators einfach aus:

Alter Wert + Speicherwert + Carry-Flag

Das neue Carry-Flag wird dann gesetzt, wenn der Wert des Akkumulators größer als 255 ist. Dann wird der Akkumulator noch auf einen Wert bis 255 zurückgesetzt (weil die Variable A, anders als der Akkumulator, Zahlen größer als 255 aufnehmen kann) und das Negativ- und das Zero-Flag, also hier die Variablen N und Z, entsprechend besetzt. Wenn jedoch das Dezimal-Flag gesetzt ist, so werden zuerst die niederwertigen 4 Bit des Akkumulators mit den niederwertigen 4 Bit des Speichers verknüpft und anschließend die höherwertigen 4 Bit. Ein Übertrag kommt immer dann zustande, wenn das Ergebnis größer als 9 ist.

Ab Zeile 10000 stehen die Unterprogrammstücke, die jeweils einem Maschinenbefehl entsprechen. Die Zeilennummern der Unterprogramme sind dabei wie folgt aufgebaut:

Zeilennummer Befehl = 10000 + 100 x Wert Maschinen-code

00	BRK	33	illegal	66	ROR Op
01	ORA (Op,X)	34	illegal	67	illegal
02	illegal	35	AND Op,X	68	PLA
03	illegal	36	ROL Op,X	69	ADC #Op
04	illegal	37	illegal	6A	ROR #Op
05	ORA Op	38	SEC	6B	illegal
06	ASL Op	39	AND Op,Y	6C	JMP (Op)
07	illegal	3A	illegal	6D	ADC Op
08	PHP	3B	illegal	6E	ROR Op
09	ORA #Op	3C	illegal	6F	illegal
0A	ASL A	3D	AND Op,X	70	BVS Op
0B	illegal	3E	ROL Op,X	71	ADC (Op),Y
0C	illegal	3F	illegal	72	illegal
0D	ORA Op	40	RTI	73	illegal
0E	ASL Op	41	EOR (Op,X)	74	illegal
0F	illegal	42	illegal	75	ADC Op,X
10	BPL Op	43	illegal	76	ROR Op,X
11	ORA (Op),Y	44	illegal	77	illegal
12	illegal	45	EOR Op	78	SEI
13	illegal	46	LSR Op	79	ADC Op,Y
14	illegal	47	illegal	7A	illegal
15	ORA Op,X	48	PHA	7B	illegal
16	ASL Op,X	49	EOR #Op	7C	illegal
17	illegal	4A	LSR A	7D	ADC Op,X
18	CLC	4B	illegal	7E	ROR Op,X
19	ORA Op,Y	4C	JMP Op	7F	illegal
1A	illegal	4D	EOR Op	80	illegal
1B	illegal	4E	LSR Op	81	STA (Op,X)
1C	illegal	4F	illegal	82	illegal
1D	ORA Op,X	50	BVC Op	83	illegal
1E	ASL Op,X	51	EOR (Op),Y	84	STY Op
1F	illegal	52	illegal	85	STA Op
20	JSR Op	53	illegal	86	STX Op
21	AND (Op,X)	54	illegal	87	illegal
22	illegal	55	EOR Op,X	88	DEY
23	illegal	56	LSR Op,X	89	illegal
24	BIT Op	57	illegal	8A	TXA
25	AND Op	58	CLI	8B	illegal
26	ROL Op	59	EOR Op,Y	8C	STY Op
27	illegal	5A	illegal	8D	STA Op
28	PLP	5B	illegal	8E	STX Op
29	AND #Op	5C	illegal	8F	illegal
2A	ROL #Op	5D	EOR Op,X	90	BCC Op
2B	illegal	5E	LSR Op,X	91	STA (Op),Y
2C	BIT Op	5F	illegal	92	illegal
2D	AND Op	60	RTS	93	illegal
2E	ROL Op	61	ADC (Op,X)	94	STY Op,X
2F	illegal	62	illegal	95	STA Op,X
30	BMI Op	63	illegal	96	STX Op,Y
31	AND (Op),Y	64	illegal	97	illegal
32	illegal	65	ADC Op	98	TYA

Illegale Opcodes

Da wir an dieser Stelle davon ausgehen, daß Basic hinlänglich bekannt ist, werden für die Unterprogramme zu den einzelnen Befehlen keine Erklärungen mehr gegeben. Vielmehr sollen gerade die Unterprogramme in Basic dazu dienen, Ihnen die Arbeitsweise der Befehle zu verdeutlichen.

Wie Sie aus Tabelle 1 ersehen können, sind nicht alle der 256 theoretisch möglichen Operations-Codes mit tatsächlichen Funktionen belegt. Sie können jedoch dem Prozessor einen solchen Operations-Code anbieten. Die Reaktion des Prozessors auf einen solchen Befehl ist durch seine interne Struktur definiert. Bei einigen illegalen Befehlen hört der Prozessor auf zu arbeiten und kann dann nur noch mit einem Reset-Impuls reaktiviert werden. Einige Befehle führen jedoch nicht zum Absturz des Computers, sondern führen exotische Funktionen aus, die eine Kombination von offiziellen Operationen darstellen. So zum Beispiel die folgenden drei Befehle:

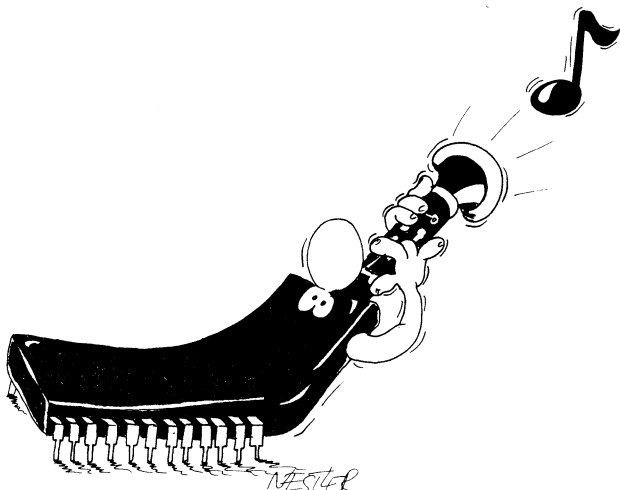
- A7 aa Der Akkumulator und das X-Register werden mit dem Inhalt der Zero-Page-Adresse aa geladen.
- 87 aa Das Ergebnis einer UND-Funktion zwischen Akku und X-Register wird in die Zero-Page-Adresse aa gespeichert.
- 97 aa Das Ergebnis einer UND-Funktion zwischen Akku und X-Register wird in die Zero-Page-Adresse aa+Y gespeichert.

Um das Programm nicht noch weiter aufzublähen, wurde beim 6502/7501-Simulator auf die Auswertung illegaler Opcodes verzichtet. (ev)

Buchauszug »Das Commodore 64 Buch«, Band 4, M&T-Verlag AG

99	STA Op,Y	BB	illegal	DD	CMP Op,X
9A	TXS	BC	LDY Op,X	DE	DEC Op,X
9B	illegal	BD	LDA Op,Y	DF	illegal
9C	illegal	BE	LDX Op,Y	E0	CPX #Op
9D	STA Op,X	BF	illegal	E1	SBC (Op,X)
9E	illegal	C0	CPY #Op	E2	illegal
9F	illegal	C1	CMP (Op,X)	E3	illegal
A0	LDY #Op	C2	illegal	E4	CPX Op
A1	LDA (Op,X)	C3	illegal	E5	SBC Op
A2	LDX #Op	C4	CPY Op	E6	INC Op
A3	illegal	C5	CMP Op	E7	illegal
A4	LDY Op	C6	DEC Op	E8	INX
A5	LDA Op	C7	illegal	E9	SBC #Op
A6	LDX Op	C8	INY	EA	NOP
A7	illegal	C9	CMP #Op	EB	illegal
A8	TAY	CA	DEX	EC	CPX Op
A9	LDA #Op	CB	illegal	ED	SBC Op
AA	TAX	CC	CPY Op	EE	INC Op
AB	illegal	CD	CMP Op	EF	illegal
AC	LDY Op	CE	DEC Op	F0	BEQ Op
AD	LDA Op	CF	illegal	F1	SBC Op,Y
AE	LDX Op	D0	BNE Op	F2	illegal
AF	illegal	D1	CMP (Op),Y	F3	illegal
B0	BCS Op	D2	illegal	F4	illegal
B1	LDA (Op),Y	D3	illegal	F5	SBC Op,X
B2	illegal	D4	illegal	F6	INC Op,X
B3	illegal	D5	CMP Op,X	F7	illegal
B4	illegal	D6	DEC Op,X	F8	SED
B5	LDA Op,X	D7	illegal	F9	SBC Op,Y
B6	LDX Op,Y	D8	CLD	FA	illegal
B7	illegal	D9	CMP Op,X	FB	illegal
B8	CLV	DA	illegal	FC	illegal
B9	LDA Op,Y	DB	illegal	FD	SBC Op,X
BA	TSX	DC	illegal	FE	INC Op,X
				FF	illegal

Tabelle 1. Die Maschinenbefehle des 6502/7501-Prozessors



DEMO6502			10 – 50000	
Variablen:				
Name	Typ	Bereich	Bedeutung	
A	G	0...255	Akkumulator	
B	G	0,1	Break-Flag	
C	G	0,1	Carry-Flag	
D	G	0,1	Dezimal-Flag	
H	H	Integer	Hilfsvariable	
H\$	H	2 Zeichen	Hilfsvariable	
HO\$	H	31 Zeichen	Eingabezeile	
HE\$	G	" 0123456789ABCDEF "	Hexadezimalziffern	
HH	H	0...65535	Hilfsvariable	
HH\$	H	4 Zeichen	Hilfsvariable	
HI	H/R	0...255	Höherwertiges Byte	
HP	H	0...10	Anzahl Abbruchpunkte	
I	G	0,1	Interrupt-Disable-Flag	
J	H	Integer	Laufvariable	
LO	H/R	0...255	Niederwertiges Byte	
M	H	0...65535	Speicheradresse	
N	G	0,1	Negativ-Flag	
O	H	0...255	Operand (1 Byte)	
OO	H	0...65535	Operand (2 Byte)	
OP	H	0...255	Befehlscode	
P	G	0...255	Prozessorstatus	
P5	G	0,1	Bit 5 von P	
PC	G	0...65535	Programmzähler	
S	G	0...255	Stapelzeiger	
V	G	0,1	Overflow-Flag	
X	G	0...255	X-Register	
Y	G	0...255	Y-Register	
Z	G	0,1	Zero-Flag	
Felder (Arrays):				
Name	Dimen.	Typ	Bereich	Bedeutung
HP	10	G	0...65535	Abbruchstellen
ST%	255	G	0...255	Stack

Tabelle 2a. Variablenliste zu »Demo 6502«

Unterprogrammaufrufe : (nur im Bereich 10 – 8440)			
in	nach	Zweck	
22	2000	Anzeige Register	
26	400	Wert von 4stelliger Hexzahl bestimmen	
44	350	Wert von 2stelliger Hexzahl bestimmen	
46	350	Wert von 2stelliger Hexzahl bestimmen	
48	350	Wert von 2stelliger Hexzahl bestimmen	
50	350	Wert von 2stelliger Hexzahl bestimmen	
62	400	Wert von 4stelliger Hexzahl bestimmen	
70	49000	Beispielprogramm »poken«	
270	200	2stellige Hexzahl bilden	
300	200	2stellige Hexzahl bilden	
420	350	Wert von 2stelliger Hexzahl bestimmen	
450	350	Wert von 2stelliger Hexzahl bestimmen	
1010	2000	Anzeige Register	
1045	200	2-stellige Hexzahl bilden	
1050	1100	Befehl ausführen	
2010	250	4stellige Hexzahl bilden	
2040	200	2stellige Hexzahl bilden	
2050	200	2stellige Hexzahl bilden	
2060	200	2stellige Hexzahl bilden	
2070	200	2stellige Hexzahl bilden	
Verzweigungen nach außen:			
In Ze	nach	Bedingung	Bemerkung
1090	STOP	PC = HP(J)	Abbruchpunkt erreicht
1370	STOP	OP größer als 255	
1410	STOP		illegaler Befehl

Tabelle 2b. Unterprogrammaufruf zu »Demo 6502«

```

10 PRINT " (CLR) (3SPACE)PC (3SPA
CE)NV-BDIZC (3SPACE)AC (2SPAC
E)XR (2SPACE)YR (2SPACE)SP      <142>
12 HE$="0123456789ABCDEF"        <255>
14 DIM ST%(255):REM STACK        <210>
16 DEF FN EO(M)=(M AND (NOT A)
) OR (NOT M) AND A                <182>
18 DEF FN V(X)=X-49+7*(X>64)       <016>
20 DEF FN F0(X)=(ASC(MID$(H0$,
X,1))-48) AND 1                   <180>
22 GOSUB 2000                      <242>
24 INPUT " (HOME) (DOWN)";H0$     <239>
26 HH$=LEFT$(H0$,4):GOSUB 400:
PC=HH                              <090>
28 N=FN F0(7)                      <181>
30 V=FN F0(8)                      <216>
32 P5=FN F0(9)                     <214>
34 B=FN F0(10)                     <055>
36 D=FN F0(11)                     <067>
38 I=FN F0(12)                     <091>
40 Z=FN F0(13)                     <163>
42 C=FN F0(14)                     <075>
44 H$=MID$(H0$,18,2):GOSUB 350
:A=H                                <254>
46 H$=MID$(H0$,22,2):GOSUB 350
:X=H                                <110>
48 H$=MID$(H0$,26,2):GOSUB 350
:Y=H                                <240>
50 H$=MID$(H0$,30,2):GOSUB 350
:S=H                                <024>
52 PRINT " (2DOWN)"                <102>
54 HP=HP+1                          <167>
56 PRINT HP" (LEFT). ABRUCHPUN
KT";                                <071>
58 INPUT " (3SPACE) (3LEFT)";HH$   <114>
60 IF HH$="X" THEN HP=HP-1:GOT
O 68                                <045>
62 GOSUB 400                        <030>
64 HP (HP)=HH                      <121>
66 GOTO 54                          <084>
68 PRINT " (UP) (23SPACE)";        <230>
70 GOSUB 49000                      <218>
72 GOTO 1000                        <012>
140 :                                <116>
150 REM *** WERT IN HIGH- UND
LOW ZERLEGEN ***                   <029>
160 HI=INT (HH/256)                 <182>
170 LO=HH-256*HI                   <246>
180 RETURN                          <238>
190 :                                <166>
200 REM * 2-STELLIGE HEXZAHL
(H$) AUS H BILDEN                  <187>
*
210 H$=MID$(HE$,H/16+1,1)+MID$
(HE$, (H AND 15)+1,1)              <166>
220 RETURN                          <022>
240 :                                <216>
250 REM * 4-STELLIGE HEXZAHL
(HH$) AUS HH BILDEN                <196>
*
260 H=INT (HH/256)                  <122>
270 GOSUB 200                       <224>
280 HH$=H$                          <057>
290 H=HH-256*H                      <255>
300 GOSUB 200                       <254>
310 HH$=HH$+H$                     <036>
320 RETURN                          <124>
340 :                                <062>
350 REM * WERT VON 2-STELLIGER
HEX-ZAHL (H$) -> H                 <084>
*
360 H=16*FN V(ASC(H$))+FN V(AS
C(MID$(H$,2)))                     <118>
370 RETURN                          <174>
390 :                                <112>
400 REM * WERT VON 4-STELLIGER
HEX-ZAHL (HH$) -> HH               <098>
*
410 H$=LEFT$(HH$,2)                <179>
420 GOSUB 350                       <206>
430 HH=256*H                        <040>
440 H$=RIGHT$(HH$,2)               <241>
450 GOSUB 350                       <236>
460 HH=HH+H                         <020>
470 RETURN                          <018>
990 :                                <204>
1000 REM *** SCHLEIFE FUER JED
EN BEFEHL ***                      <011>
1010 GOSUB 2000                     <212>
1020 IF HP=0 THEN 1000              <126>
1030 FOR J=1 TO HP:IF PC=HP(J)
THEN 1080                          <024>
1035 NEXT J                         <111>
1040 OP=PEEK(PC)                    <033>
1045 H=OP:GOSUB 200:PRINT "OPC
ODE: "H$" (2SPACE)";              <153>
1050 GOSUB 1100                     <006>
1055 PC=PC+1                        <013>
1060 GOTO 1000                      <240>
1080 PRINT "ABBRUCH ("J;";" );    <150>
1090 STOP                           <140>
1100 IF OP=0 THEN 10000             <011>
1110 ON OP GOTO 10100, 1400
, 1400, 1400, 10500, 10600,

```

Listing »Demo 6502«

1120	1400,10800,10900,11000	<136>	4020	O=PEEK(PC)	<179>	10660	POKE O,M	<057>
1130	ON OP-10 GOTO 1400, 1400		4030	PC=PC+1	<194>	10670	Z=-(M=0)	<235>
	,11300,11400, 1400,11600,	<021>	4040	OO=D+256*PEEK(PC)	<205>	10680	N=(M AND 128)/128	<160>
	11700, 1400, 1400, 1400		4050	RETURN	<042>	10690	RETURN	<078>
1130	ON OP-20 GOTO 12100,12200		4990	:	<140>	10700	:	<098>
	, 1400,12400,12500, 1400,		5000	REM *** STATUS ZUSAMMENFA	<176>	10800	REM *** PHP	<130>
	1400, 1400,12900,13000	<128>		SSEN ***		10810	GOSUB 5000	<130>
1140	ON OP-30 GOTO 1400,13200		5010	P=128*N+64*V+32*P+16*B+8	<211>	10820	ST%(S)=P:S=(S-1) AND 255	<190>
	,13300, 1400, 1400,13600,	<091>		*D+4*I+2*Z+C	<252>	10830	RETURN	<220>
	13700,13800, 1400,14000		5020	RETURN	<124>	10890	:	<198>
1150	ON OP-40 GOTO 14100,14200		5990	:		10900	REM *** ORA #IMMEDIATE	<159>
	, 1400,14400,14500,14600,	<082>	6000	REM *** STATUS ZERLEGEN *	<188>	10910	GOSUB 3000	<214>
	1400,14800,14900, 1400		**			10920	A=A OR 0	<147>
1160	ON OP-50 GOTO 1400, 1400		6010	N=(P AND 128)/128	<159>	10930	N=(A AND 128)/128	<027>
	,15300,15400, 1400,15600,	<046>	6020	V=(P AND 64)/64	<098>	10940	Z=-(A=0)	<248>
	15700, 1400, 1400, 1400		6030	P5=(P AND 32)/32	<226>	10950	RETURN	<084>
1170	ON OP-60 GOTO 16100,16200		6040	B=(P AND 16)/16	<157>	10990	:	<042>
	, 1400,16400,16500, 1400,	<038>	6050	D=(P AND 8)/8	<109>	11000	REM *** ASL A	<137>
	1400, 1400,16900,17000		6060	I=(P AND 4)/4	<121>	11010	A=A*2	<039>
1180	ON OP-70 GOTO 1400,17200		6070	Z=(P AND 2)/2	<190>	11020	C=-(A>255)	<164>
	,17300,17400, 1400,17600,	<088>	6080	C=P AND 1	<214>	11030	A=A AND 255	<106>
	17700,17800, 1400,18000		6090	RETURN	<050>	11040	Z=-(A=0)	<157>
1190	ON OP-80 GOTO 18100, 1400		6990	:	<108>	11050	N=(A AND 128)/128	<149>
	, 1400, 1400,18500,18600,	<246>	7000	REM *** RELATIVEN SPRUNG	<123>	11060	RETURN	<196>
	1400,18800,18900, 1400		AUSFUEHREN ***			11290	:	<090>
1200	ON OP-90 GOTO 1400, 1400		7010	IF D>127 THEN D=0-256	<153>	11300	REM *** ORA ABSOLUTE	<149>
	,19300,19400, 1400,19600,	<072>	7020	PC=PC+O	<166>	11310	GOSUB 4000	<114>
	19700, 1400, 1400, 1400		7030	RETURN	<230>	11320	A=A OR PEEK(OO)	<169>
1210	ON OP-100GOTO 20100,20200		7990	:	<092>	11330	N=(A AND 128)/128	<175>
	, 1400,20400,20500,20600,	<237>	8000	REM *** A=A+M+C (ADC-BEFE	<060>	11340	Z=-(A=0)	<140>
	1400,20800,20900,21000		HL) ***			11350	RETURN	<232>
1220	ON OP-110GOTO 1400,21200		8010	IF D=0 THEN A=A+M+C:C=-(A	<143>	11390	:	<190>
	,21300, 1400, 1400, 1400	<018>		>255):GOTO 8070		11400	REM *** ASL ABSOLUTE	<001>
	21700,21800, 1400,22000		8020	H=(A AND 15)+(M AND 15)+C	<033>	11410	GOSUB 4000	<214>
1230	ON OP-120GOTO 22100, 1400		8030	C=-(H>9)	<114>	11420	M=PEEK(OO)	<146>
	, 1400, 1400,22500,22600,	<053>	8040	H=H AND 15	<187>	11430	M=M*2	<189>
	1400, 1400,22900, 1400		8050	A=(A AND 240)+(M AND 240)	<179>	11440	C=-(M>255)	<077>
1240	ON OP-130GOTO 1400,23200			+H+16*C		11450	M=M AND 255	<000>
	,23300,23400, 1400,23600,	<066>	8060	C=-(A>99)	<146>	11460	POKE OO,M	<192>
	1400,23800, 1400,24000		8070	A=A AND 255	<194>	11470	Z=-(M=0)	<017>
1250	ON OP-140GOTO 24100,24200		8080	N=(A AND 128)/128	<227>	11480	N=(M AND 128)/128	<198>
	, 1400,24400,24500, 1400,	<089>	8090	Z=-(A=0)	<192>	11490	RETURN	<116>
	1400,24800,24900,25000		8100	RETURN	<028>	11590	:	<136>
1260	ON OP-150GOTO 1400,25200		8190	:	<036>	11600	REM *** BPL	<036>
	,25300,25400, 1400, 1400,	<165>	8200	REM *** A=A-M-1+C (SBC-BE	<002>	11610	GOSUB 3000	<152>
	25700, 1400, 1400,26000		FEHL) ***			11620	IF N=0 THEN GOSUB 7000	<022>
1270	ON OP-160GOTO 26100,26200		8210	IF D=0 THEN A=A-M-1+C:C=-(	<044>	11630	RETURN	<002>
	, 1400,26400,26500,26600,	<037>		A>=0):GOTO 8270		11690	:	<236>
	1400,26800,26900,27000		8220	H=(A AND 15)-(M AND 15)-1	<002>	11700	REM ** ORA (INDIRECT),Y	<097>
1280	ON OP-170GOTO 1400,27200			+C		11710	GOSUB 3000	<252>
	,27300,27400, 1400,27600,	<055>	8230	C=-(H>=0)	<165>	11720	OO=PEEK(O)+256*PEEK(O+1)	<062>
	27700, 1400, 1400,28000		8240	H=H AND 15	<057>		+Y	
1290	ON OP-180GOTO 28100,28200		8250	A=(A AND 240)-(M AND 240)	<239>	11730	A=A OR PEEK(OO)	<069>
	, 1400,28400,28500,28600,	<143>		+H-16+16*C		11740	N=(A AND 128)/128	<075>
	1400,28800,28900,29000		8260	C=-(A>=0)	<226>	11750	Z=-(A=0)	<040>
1300	ON OP-190GOTO 1400,29200		8270	A=A AND 255	<203>	11760	RETURN	<132>
	,29300, 1400, 1400,29600,	<242>	8280	C=(A AND 128)/128	<074>	12090	:	<128>
	29700,29800, 1400,30000		8290	Z=-(A=0)	<212>	12100	REM *** ORA ZEROPAGE,X	<159>
1310	ON OP-200GOTO 30100,30200		8300	RETURN	<137>	12110	GOSUB 3000	<144>
	, 1400,30400,30500,30600,	<169>	8390	:	<058>	12120	A=A OR PEEK(O+X)	<140>
	1400,30800,30900, 1400		8400	REM *** A-M (CMP-BEFEHL)	<086>	12130	N=(A AND 128)/128	<213>
1320	ON OP-210GOTO 1400, 1400		**			12140	Z=-(A=0)	<178>
	,31300,31400, 1400,31600,	<104>	8410	C=-(A-M)>=0	<085>	12150	RETURN	<014>
	31700, 1400, 1400, 1400		8420	N=-(A-M AND 128)/128	<239>	12190	:	<228>
1330	ON OP-220GOTO 32100,32200		8430	Z=-(A-M)=0	<241>	12200	REM *** ASL ZEROPAGE,X	<011>
	, 1400,32400,32500, 1400,	<212>	8440	RETURN	<021>	12210	GOSUB 3000	<244>
	1400,32800,32900,33000		9990	:	<060>	12220	M=PEEK(O+X)	<231>
1340	ON OP-230GOTO 1400,33200		10000	REM *** BRK	<216>	12230	M=M*2	<227>
	,33300,33400, 1400,33600,	<101>	10050	HH=PC+2:GOSUB 150	<222>	12240	C=-(M>255)	<115>
	33700,33800, 1400,34000		10010	ST%(S)=HI:S=(S-1) AND 25	<208>	12250	M=M AND 255	<038>
1350	ON OP-240GOTO 34100, 1400			5	<245>	12260	POKE O+X,M	<128>
	, 1400, 1400,34500,34600,	<224>	10015	ST%(S)=LO:S=(S-1) AND 25	<035>	12270	Z=-(M=0)	<055>
	1400,34800,34900, 1400			5		12280	N=(M AND 128)/128	<236>
1360	ON OP-250GOTO 1400, 1400		10020	B=1	<107>	12290	RETURN	<156>
	,35300,35400, 1400	<194>	10025	GOSUB 5000	<162>	12390	:	<174>
1370	STOP	<166>	10030	ST%(S)=P:S=(S-1) AND 255	<078>	12400	REM *** CLC	<127>
1400	PRINT "UNGUELTIGER CODE";	<011>	10035	I=1		12410	C=0	<127>
1410	STOP	<206>	10040	PC=PEEK(65534)+256*PEEK(	<243>	12420	RETURN	<030>
1990	:	<188>		65535)		12490	:	<018>
2000	REM *** ANZEIGE ***	<070>	10045	GOSUB 2000	<103>	12500	REM *** ORA ABSOLUTE,Y	<010>
2010	HH=PC:GOSUB 250:PC=HH	<198>	10050	PRINT "BRK ERREICHT";	<220>	12510	GOSUB 4000	<042>
2020	ST%=CHR\$(48+N)+CHR\$(48+V)		10055	END	<151>	12520	A=A OR PEEK(OO+Y)	<195>
	+CHR\$(48+P5)+CHR\$(48+B)	<249>	10090	:	<160>	12530	N=(A AND 128)/128	<103>
2030	ST%=ST%+CHR\$(48+D)+CHR\$(4		10100	REM *** ORA (INDIRECT,X)	<238>	12540	Z=-(A=0)	<068>
	B+I)+CHR\$(48+Z)+CHR\$(48+C	<129>	10110	GOSUB 3000	<176>	12550	RETURN	<162>
	)		10120	OO=PEEK(O+X)+256*PEEK(O+	<030>	12890	:	<166>
2040	H=A:GOSUB 200:A=H\$	<209>		X+1)		12900	REM *** ORA ABSOLUTE,X	<154>
2050	H=X:GOSUB 200:X=H\$	<006>	10130	A=A OR PEEK(OO)	<249>	12910	GOSUB 4000	<190>
2060	H=Y:GOSUB 200:Y=H\$	<040>	10140	N=(A AND 128)/128	<255>	12920	A=A OR PEEK(OO+X)	<079>
2070	H=S:GOSUB 200:S=H\$	<162>	10150	Z=-(A=0)	<220>	12930	N=(A AND 128)/128	<251>
2090	PRINT " (HOME) (DOWN) (2SPAC		10160	RETURN	<056>	12940	Z=-(A=0)	<216>
	E)"PC\$((2SPACE)"ST\$((3SPA	<071>	10490	:	<050>	12950	RETURN	<052>
	CE)"A\$((2SPACE)"X\$((2SPAC		10500	REM *** ORA ZEROPAGE	<154>	12990	:	<010>
	E)"Y\$((2SPACE)"S\$	<126>	10510	GOSUB 3000	<068>	13000	REM *** ASL ABSOLUTE,X	<006>
2100	RETURN	<172>	10520	A=A OR PEEK(O)	<147>	13010	GOSUB 4000	<034>
2990	:		10530	N=(A AND 128)/128	<137>	13020	M=PEEK(OO+X)	<179>
3000	REM *** 1-BYTE-OPERAND HO	<242>	10540	Z=-(A=0)	<102>	13030	M=M*2	<009>
	LEN ***		10550	RETURN	<194>	13040	C=-(M>255)	<153>
3010	PC=PC+1	<190>	10590	:	<152>	13050	M=M AND 255	<076>
3020	O=PEEK(PC)	<195>	10600	REM *** ASL ZEROPAGE	<006>	13060	POKE OO+X,M AND 255	<039>
3030	RETURN	<038>	10610	GOSUB 3000	<168>	13070	Z=-(M=0)	<095>
3990	:	<156>	10620	M=PEEK(O)	<048>	13080	N=(M AND 128)/128	<020>
4000	REM *** 2-BYTE-OPERAND HO	<034>	10630	M=M*2	<151>	13090	RETURN	<194>
	LEN ***		10640	C=-(M>255)	<039>	13190	:	<212>
4010	PC=PC+1	<174>	10650	M=M AND 255	<218>	13200	REM *** JSR	<249>

13210 HH=PC+2:GOSUB 150	<123>	15300 REM *** AND ZEROPAGE,X	<181>	17690 :	<140>
13220 ST%(S)=HI:S=(S-1) AND 25		15310 GOSUB 3000	<040>	17700 REM *** EOR ABSOLUTE	<212>
5	<114>	15320 A=A AND PEEK(O+X)	<003>	17710 GOSUB 4000	<164>
13230 ST%(S)=LO:S=(S-1) AND 25		15330 N=(A AND 128)/128	<109>	17720 A=FN EO(PEEK(OO))	<197>
5	<156>	15340 Z--(A=0)	<074>	17730 N=(A AND 128)/128	<225>
13240 GOSUB 4000	<010>	15350 RETURN	<166>	17740 Z--(A=0)	<190>
13250 PC=OO-1	<112>	15390 :	<126>	17750 RETURN	<026>
13260 RETURN	<108>	15400 REM ***ROL ZEROPAGE,X	<231>	17790 :	<240>
13290 :	<056>	15410 GOSUB 3000	<142>	17800 REM *** LSR ABSOLUTE	<252>
13300 REM *** AND (INDIRECT,X)	<004>	15420 M=PEEK(O+X)	<129>	17810 GOSUB 4000	<008>
13310 GOSUB 3000	<072>	15430 M=M*2+C	<021>	17820 M=PEEK(OO)	<196>
13320 OO=PEEK(O+X)+256*PEEK(O+X+1)	<184>	15440 C--(M>255)	<013>	17830 C=M AND 1	<231>
13330 A=A AND PEEK(OO)	<114>	15450 M=M AND 255	<192>	17840 M=INT(M/2)	<017>
13340 N=(A AND 128)/128	<153>	15460 POKE O+X,M AND 255	<170>	17850 POKE OO,M	<232>
13350 Z--(A=0)	<118>	15470 Z--(M=0)	<209>	17860 Z--(M=0)	<057>
13360 RETURN	<210>	15480 N=(M AND 128)/128	<134>	17870 N=(M AND 128)/128	<238>
13590 :	<104>	15490 RETURN	<052>	17880 RETURN	<156>
13600 REM *** BIT ZEROPAGE	<250>	15590 :	<070>	17990 :	<186>
13610 GOSUB 3000	<120>	15600 REM *** SEC	<151>	18000 REM *** BVC	<080>
13620 M=PEEK(O)	<000>	15610 C=1	<039>	18010 GOSUB 3000	<202>
13630 N=(M AND 128)/128	<062>	15620 RETURN	<184>	18020 IF V=0 THEN GOSUB 7000	<136>
13640 V=(M AND 64)/64	<001>	15690 :	<172>	18030 RETURN	<052>
13650 Z--(A AND M)=0	<205>	15700 REM *** AND ABSOLUTE,Y	<034>	18090 :	<030>
13660 RETURN	<000>	15710 GOSUB 4000	<196>	18100 REM *** EOR (INDIRECT),Y	<015>
13690 :	<204>	15720 A=A AND PEEK(OO+Y)	<060>	18110 GOSUB 3000	<046>
13700 REM *** AND ZEROPAGE	<176>	15730 N=(A AND 128)/128	<001>	18120 OO=PEEK(O)+256*PEEK(O+1)+Y	<112>
13710 GOSUB 3000	<220>	15740 Z--(A=0)	<222>	18130 A=FN EO(PEEK(OO))	<097>
13720 A=A AND PEEK(O)	<010>	15750 RETURN	<058>	18140 N=(A AND 128)/128	<125>
13730 N=(A AND 128)/128	<033>	16090 :	<062>	18150 Z--(A=0)	<090>
13740 Z--(A=0)	<254>	16100 REM *** AND ABSOLUTE,X	<176>	18160 RETURN	<182>
13750 RETURN	<090>	16110 GOSUB 4000	<086>	18490 :	<178>
13790 :	<048>	16120 A=A AND PEEK(OO+Y)	<206>	18500 REM *** EOR ZEROPAGE,X	<222>
13800 REM *** ROL ZEROPAGE	<224>	16130 N=(A AND 128)/128	<149>	18510 GOSUB 3000	<194>
13810 GOSUB 3000	<064>	16140 Z--(A=0)	<114>	18520 A=FN EO(PEEK(O+X))	<003>
13820 M=PEEK(O)	<200>	16150 RETURN	<164>	18530 N=(A AND 128)/128	<007>
13830 M=M*2+C	<201>	16190 :	<164>	18540 Z--(A=0)	<228>
13840 C--(M>255)	<193>	16200 REM *** ROL ABSOLUTE,X	<226>	18550 RETURN	<064>
13850 M=M AND 255	<116>	16210 GOSUB 4000	<188>	18590 :	<022>
13860 POKE O,M AND 255	<118>	16220 M=PEEK(OO+X)	<077>	18600 REM *** LSR ZEROPAGE,X	<006>
13870 Z--(M=0)	<133>	16230 M=M*2+C	<059>	18610 GOSUB 3000	<038>
13880 N=(M AND 128)/128	<058>	16240 C--(M>255)	<051>	18620 M=PEEK(O+X)	<025>
13890 RETURN	<232>	16250 M=M AND 255	<230>	18630 C=M AND 1	<013>
13990 :	<250>	16260 POKE OO+X,M AND 255	<191>	18640 M=INT(M/2)	<055>
14000 REM *** PLP	<028>	16270 Z--(M=0)	<247>	18650 POKE O+X,M	<168>
14010 S=(S+1) AND 255	<130>	16280 N=(M AND 128)/128	<172>	18660 Z--(M=0)	<095>
14020 P=ST%(S)	<199>	16290 RETURN	<090>	18670 N=(M AND 128)/128	<020>
14030 GOSUB 6000	<054>	16390 :	<110>	18680 RETURN	<194>
14040 RETURN	<126>	16400 REM *** RTI	<013>	18790 :	<224>
14090 :	<096>	16410 S=(S+1) AND 255	<246>	18800 REM *** CLI	<183>
14100 REM *** AND #IMMEDIATE	<183>	16420 P=ST%(S)	<059>	18810 I=0	<201>
14110 GOSUB 3000	<112>	16430 GOSUB 6000	<170>	18820 RETURN	<080>
14120 A=A AND O	<012>	16440 S=(S+1) AND 255	<020>	18890 :	<068>
14130 N=(A AND 128)/128	<181>	16450 PC=ST%(S)	<203>	18900 REM *** EOR ABSOLUTE,Y	<073>
14140 Z--(A=0)	<146>	16460 S=(S+1) AND 255	<040>	18910 GOSUB 4000	<092>
14150 RETURN	<238>	16470 PC=256*PC+ST%(S)	<121>	18920 A=FN EO(PEEK(OO+Y))	<121>
14190 :	<196>	16480 RETURN	<026>	18930 N=(A AND 128)/128	<153>
14200 REM *** ROL A	<101>	16490 :	<210>	18940 Z--(A=0)	<118>
14210 A=A*2+C	<087>	16500 REM *** EOR (INDIRECT,X)	<045>	18950 RETURN	<212>
14220 C--(A>255)	<060>	16510 GOSUB 3000	<226>	19290 :	<216>
14230 A=A AND 255	<002>	16520 OO=PEEK(O+X)+256*PEEK(O+X+1)	<080>	19300 REM *** EOR ABSOLUTE,X	<217>
14240 Z--(A=0)	<246>	16530 A=FN EO(PEEK(OO))	<021>	19310 GOSUB 4000	<240>
14250 N=(A AND 128)/128	<045>	16540 N=(A AND 128)/128	<049>	19320 A=FN EO(PEEK(OO+X))	<237>
14260 RETURN	<092>	16550 Z--(A=0)	<014>	19330 N=(A AND 128)/128	<045>
14390 :	<142>	16560 RETURN	<106>	19340 Z--(A=0)	<010>
14400 REM *** BIT ABSOLUTE	<020>	16890 :	<100>	19350 RETURN	<102>
14410 GOSUB 4000	<166>	16900 REM *** EOR ZEROPAGE	<217>	19390 :	<060>
14420 M=PEEK(OO)	<098>	16910 GOSUB 3000	<118>	19400 REM *** LSR ABSOLUTE,X	<001>
14430 N=(M AND 128)/128	<100>	16920 A=FN EO(PEEK(O))	<073>	19410 GOSUB 4000	<084>
14440 V=(M AND 64)/64	<039>	16930 N=(A AND 128)/128	<187>	19420 M=PEEK(OO+X)	<229>
14450 Z--(A AND M)=0	<243>	16940 Z--(A=0)	<152>	19430 C=M AND 1	<051>
14460 RETURN	<038>	16950 RETURN	<244>	19440 M=INT(M/2)	<093>
14490 :	<242>	16990 :	<202>	19450 POKE OO+X,M	<042>
14500 REM *** AND ABSOLUTE	<171>	17000 REM *** LSR ZEROPAGE	<001>	19460 Z--(M=0)	<135>
14510 GOSUB 4000	<010>	17010 GOSUB 3000	<218>	19470 N=(M AND 128)/128	<060>
14520 A=A AND PEEK(OO)	<032>	17020 M=PEEK(O)	<098>	19480 RETURN	<234>
14530 N=(A AND 128)/128	<071>	17030 C=M AND 1	<193>	19590 :	<006>
14540 Z--(A=0)	<036>	17040 M=INT(M/2)	<235>	19600 REM *** RTS	<175>
14550 RETURN	<128>	17050 POKE O,M	<097>	19610 S=(S+1) AND 255:HI=ST%(S)	
14590 :	<086>	17060 Z--(M=0)	<019>		<147>
14600 REM *** ROL ABSOLUTE	<221>	17070 N=(M AND 128)/128	<200>	19620 S=(S+1) AND 255:LO=ST%(S)	
14610 GOSUB 4000	<112>	17080 RETURN	<118>		<161>
14620 M=PEEK(OO)	<044>	17190 :	<148>	19630 PC=256*HI+LO	<221>
14630 M=M*2+C	<239>	17200 REM *** PHA	<165>	19640 REM PC=PC+1 GESCHIEHT B	
14640 C--(M>255)	<231>	17210 ST%(S)=A:S=(S-1) AND 255	<200>	EI 1040	<116>
14650 M=M AND 255	<154>	17220 RETURN	<004>	19650 RETURN	<148>
14660 POKE OO,M AND 255	<161>	17290 :	<248>	19690 :	<106>
14670 Z--(M=0)	<171>	17300 REM *** EOR #IMMEDIATE	<222>	19700 REM *** ADC (INDIRECT,X)	<048>
14680 N=(M AND 128)/128	<096>	17310 GOSUB 3000	<008>	19710 GOSUB 3000	<122>
14690 RETURN	<014>	17320 A=FN EO(O)	<082>	19720 OO=PEEK(O+X)+256*PEEK(O+X+1)	<234>
14790 :	<032>	17330 N=(A AND 128)/128	<077>	19730 M=PEEK(OO)	<074>
14800 REM *** BMI	<055>	17340 Z--(A=0)	<042>	19740 GOSUB 8000	<194>
14810 GOSUB 3000	<048>	17350 RETURN	<134>	19750 RETURN	<250>
14820 IF N=1 THEN GOSUB 7000	<206>	17390 :	<092>	20090 :	<254>
14830 RETURN	<154>	17400 REM *** LSR A	<132>	20100 REM *** ADC ZEROPAGE	<220>
14890 :	<134>	17410 C=A AND 1	<129>	20110 GOSUB 3000	<014>
14900 REM *** AND (INDIRECT),Y	<232>	17420 A=INT(A/2)	<056>	20120 M=PEEK(O)	<150>
14910 GOSUB 3000	<150>	17430 Z--(A=0)	<134>	20130 GOSUB 8000	<074>
14920 OO=PEEK(O)+256*PEEK(O+1)+Y	<216>	17440 N=(A AND 128)/128	<076>	20140 RETURN	<130>
14930 A=A AND PEEK(OO)	<190>	17450 RETURN	<236>	20190 :	<098>
14940 N=(A AND 128)/128	<229>	17590 :	<038>	20200 REM *** ROR ZEROPAGE	<024>
14950 Z--(A=0)	<194>	17600 REM *** JMP ABSOLUTE	<172>	20210 GOSUB 3000	<114>
14960 RETURN	<030>	17610 GOSUB 4000	<062>		
15290 :	<024>	17620 PC=OO-1	<164>		
		17630 RETURN	<160>		

Listing »Demo 6502« (Fortsetzung)

```

20220 M=PEEK(0) <250>
20230 M=(128*C)+M/2 <254>
20240 C=-(M<>INT(M)) <024>
20250 M=M AND 255 <166>
20260 POKE 0,M <005>
20270 Z=-(M=0) <183>
20280 N=(M AND 128)/128 <108>
20290 RETURN <026>
20390 : <044>
20400 REM *** PHP <076>
20410 S=(S+1) AND 255 <180>
20420 A=ST%(S) <189>
20430 RETURN <166>
20490 : <146>
20500 REM *** ADC #IMMEDIATE <227>
20510 GOSUB 3000 <162>
20520 M=0 <134>
20530 GOSUB 8000 <222>
20540 RETURN <022>
20590 : <246>
20600 REM *** ROR A <157>
20610 A=(128*C)+A/2 <140>
20620 C=-(A<>INT(A)) <051>
20630 A=A AND 255 <052>
20640 Z=-(A=0) <040>
20650 N=(A AND 128)/128 <095>
20660 RETURN <142>
20790 : <192>
20800 REM *** JMP INDIRECT <241>
20810 GOSUB 4000 <216>
20820 PC=PEEK(00)+256*PEEK(00+1)-1 <024>
20830 RETURN <058>
20890 : <036>
20900 REM *** ADC ABSOLUTE <215>
20910 GOSUB 4000 <060>
20920 M=PEEK(00) <248>
20930 GOSUB 8000 <112>
20940 RETURN <168>
20990 : <136>
21000 REM *** ROR ABSOLUTE <118>
21010 GOSUB 4000 <162>
21020 M=PEEK(00) <094>
21030 M=(128*C)+M/2 <036>
21040 C=-(M<>INT(M)) <062>
21050 M=M AND 255 <204>
21060 POKE 0,M <140>
21070 Z=-(M=0) <221>
21080 N=(M AND 128)/128 <146>
21090 RETURN <064>
21190 : <082>
21200 REM *** BVS <248>
21210 GOSUB 3000 <098>
21220 IF V=1 THEN GOSUB 7000 <064>
21230 RETURN <204>
21290 : <184>
21300 REM *** ADC (INDIRECT),Y <020>
21310 GOSUB 3000 <200>
21320 00=PEEK(0)+256*PEEK(0+1)+Y <010>
21330 M=PEEK(00) <150>
21340 GOSUB 8000 <014>
21350 RETURN <070>
21690 : <074>
21700 REM *** ADC ZEROPAGE,X <225>
21710 GOSUB 3000 <090>
21720 M=PEEK(0+X) <077>
21730 GOSUB 8000 <150>
21740 RETURN <206>
21790 : <176>
21800 REM *** ROR ZEROPAGE,X <031>
21810 GOSUB 3000 <192>
21820 M=PEEK(0+X) <179>
21830 M=(128*C)+M/2 <074>
21840 C=-(M<>INT(M)) <100>
21850 M=M AND 255 <242>
21860 POKE 0+X,M <076>
21870 Z=-(M=0) <003>
21880 N=(M AND 128)/128 <184>
21890 RETURN <102>
21990 : <120>
22000 REM *** SEI <207>
22010 I=1 <113>
22020 RETURN <234>
22090 : <222>
22100 REM *** ADC ABSOLUTE,Y <078>
22110 GOSUB 4000 <246>
22120 M=PEEK(00+Y) <137>
22130 GOSUB 8000 <042>
22140 RETURN <098>
22490 : <112>
22500 REM *** ADC ABSOLUTE,X <220>
22510 GOSUB 4000 <136>
22520 M=PEEK(00+X) <025>
22530 GOSUB 8000 <190>
22540 RETURN <246>
22590 : <214>
22600 REM *** ROR ABSOLUTE,X <026>
22610 GOSUB 4000 <238>
22620 M=PEEK(00+X) <127>
22630 M=(128*C)+M/2 <112>
22640 C=-(M<>INT(M)) <138>
22650 M=M AND 255 <024>
22660 POKE 00+X,M AND 255 <241>
22670 Z=-(M=0) <041>

```

```

22680 N=(M AND 128)/128 <222>
22690 RETURN <140>
22890 : <004>
22900 REM *** STA (INDIRECT,X) <084>
22910 GOSUB 3000 <020>
22920 00=PEEK(0+X)+256*PEEK(0+X+1) <130>
22930 POKE 00,A <229>
22940 RETURN <136>
23190 : <050>
23200 REM *** STY ZEROPAGE <178>
23210 GOSUB 3000 <066>
23220 POKE 0,Y <045>
23230 RETURN <172>
23290 : <150>
23300 REM *** STA ZEROPAGE <000>
23310 GOSUB 3000 <168>
23320 POKE 0,A <144>
23330 RETURN <018>
23390 : <252>
23400 REM *** STX ZEROPAGE <123>
23410 GOSUB 3000 <012>
23420 POKE 0,X <215>
23430 RETURN <118>
23590 : <198>
23600 REM *** DEY <106>
23610 Y=(Y-1) AND 255 <103>
23620 Z=-(Y=0) <234>
23630 N=(Y AND 128)/128 <030>
23640 RETURN <074>
23790 : <142>
23800 REM *** TXA <168>
23810 A=X <219>
23820 Z=-(A=0) <174>
23830 N=(A AND 128)/128 <229>
23840 RETURN <122>
23990 : <088>
24000 REM *** STY ABSOLUTE <173>
24010 GOSUB 4000 <112>
24020 POKE 00,Y <053>
24030 RETURN <210>
24090 : <190>
24100 REM *** STA ABSOLUTE <251>
24110 GOSUB 4000 <214>
24120 POKE 00,A <149>
24130 RETURN <056>
24190 : <034>
24200 REM *** STX ABSOLUTE <118>
24210 GOSUB 4000 <058>
24220 POKE 00,X <191>
24230 RETURN <156>
24390 : <236>
24400 REM *** BCC <248>
24410 GOSUB 3000 <252>
24420 IF C=0 THEN GOSUB 7000 <034>
24430 RETURN <102>
24490 : <080>
24500 REM *** STA (INDIRECT),Y <054>
24510 GOSUB 3000 <096>
24520 00=PEEK(0)+256*PEEK(0+1)+Y <162>
24530 POKE 00,A <049>
24540 RETURN <212>
24790 : <126>
24800 REM *** STY ZEROPAGE,X <183>
24810 GOSUB 3000 <142>
24820 POKE 0+X,Y <248>
24830 RETURN <228>
24890 : <005>
24900 REM *** STA ZEROPAGE,X <244>
24910 GOSUB 3000 <082>
24920 POKE 0+X,A <094>
24930 RETURN <072>
24990 : <072>
25000 REM *** STX ZEROPAGE,Y <132>
25010 GOSUB 3000 <088>
25020 POKE 0+Y,X <098>
25030 RETURN <194>
25190 : <018>
25200 REM *** TYA <172>
25210 A=Y <109>
25220 Z=-(A=0) <048>
25230 N=(A AND 128)/128 <103>
25240 RETURN <150>
25290 : <118>
25300 REM *** STA ABSOLUTE,Y <112>
25310 GOSUB 4000 <142>
25320 POKE 00+Y,A <122>
25330 RETURN <240>
25390 : <220>
25400 REM *** TXS <008>
25410 S=X <111>
25420 RETURN <076>
25490 : <010>
25700 REM *** STA ABSOLUTE,X <000>
25710 GOSUB 4000 <034>
25720 POKE 00+X,A <206>
25730 RETURN <132>
25990 : <056>
26000 REM *** LDY #IMMEDIATE <098>
26010 GOSUB 3000 <072>
26020 Y=0 <092>
26030 N=(Y AND 128)/128 <144>
26040 Z=-(Y=0) <112>
26050 RETURN <198>

```

```

26090 : <156>
26100 REM *** LDA (INDIRECT,X) <035>
26110 GOSUB 3000 <172>
26120 00=PEEK(0+X)+256*PEEK(0+X+1) <028>
26130 A=PEEK(00) <076>
26140 N=(A AND 128)/128 <253>
26150 Z=-(A=0) <218>
26160 RETURN <054>
26190 : <002>
26200 REM *** LDX #IMMEDIATE <043>
26210 GOSUB 3000 <018>
26220 X=0 <034>
26230 N=(X AND 128)/128 <058>
26240 Z=-(X=0) <250>
26250 RETURN <144>
26390 : <204>
26400 REM *** LDY ZEROPAGE <174>
26410 GOSUB 3000 <220>
26420 Y=PEEK(0) <148>
26430 N=(Y AND 128)/128 <036>
26440 Z=-(Y=0) <084>
26450 RETURN <090>
26490 : <048>
26500 REM ** LDA ZEROPAGE <004>
26510 GOSUB 3000 <064>
26520 A=PEEK(0) <152>
26530 N=(A AND 128)/128 <133>
26540 Z=-(A=0) <098>
26550 RETURN <190>
26590 : <148>
26600 REM *** LDX ZEROPAGE <074>
26610 GOSUB 3000 <164>
26620 X=PEEK(0) <088>
26630 N=(X AND 128)/128 <206>
26640 Z=-(X=0) <142>
26650 RETURN <036>
26790 : <094>
26800 REM *** TAY <084>
26810 Y=A <152>
26820 Z=-(Y=0) <130>
26830 N=(Y AND 128)/128 <182>
26840 RETURN <226>
26890 : <196>
26900 REM *** LDA #IMMEDIATE <214>
26910 GOSUB 3000 <212>
26920 A=0 <136>
26930 N=(A AND 128)/128 <025>
26940 Z=-(A=0) <246>
26950 RETURN <082>
26990 : <040>
27000 REM *** TAX <205>
27010 X=A <094>
27020 Z=-(X=0) <012>
27030 N=(X AND 128)/128 <096>
27040 RETURN <172>
27190 : <242>
27200 REM *** LDY ABSOLUTE <126>
27210 GOSUB 4000 <010>
27220 Y=PEEK(00) <246>
27230 N=(Y AND 128)/128 <074>
27240 Z=-(Y=0) <042>
27250 RETURN <128>
27290 : <086>
27300 REM *** LDA ABSOLUTE <202>
27310 GOSUB 4000 <110>
27320 A=PEEK(00) <250>
27330 N=(A AND 128)/128 <171>
27340 Z=-(A=0) <136>
27350 RETURN <228>
27390 : <186>
27400 REM *** LDX ABSOLUTE <071>
27410 GOSUB 4000 <212>
27420 X=PEEK(00) <188>
27430 N=(X AND 128)/128 <244>
27440 Z=-(X=0) <180>
27450 RETURN <074>
27590 : <132>
27600 REM *** CLV <104>
27610 V=0 <161>
27620 RETURN <244>
27690 : <234>
27700 REM *** LDA (INDIRECT),Y <007>
27710 GOSUB 3000 <250>
27720 00=PEEK(0)+256*PEEK(0+1)+Y <060>
27730 A=PEEK(00) <152>
27740 N=(A AND 128)/128 <073>
27750 Z=-(A=0) <038>
27760 RETURN <130>
27990 : <024>
28000 REM *** LDY ZEROPAGE,X <136>
28010 GOSUB 3000 <040>
28020 Y=PEEK(0+X) <075>
28030 N=(Y AND 128)/128 <112>
28040 Z=-(Y=0) <080>
28050 RETURN <166>
28090 : <124>
28100 REM *** LDA ZEROPAGE,X <212>
28110 GOSUB 3000 <140>
28120 A=PEEK(0+X) <079>
28130 N=(A AND 128)/128 <209>
28140 Z=-(A=0) <174>
28150 RETURN <010>
28190 : <226>

```

28200 REM *** LDX ZEROPAGE,X	<081>	30450 Z--((Y-M)=0)	<247>	33050 Z--(M=0)	<008>
28210 GOSUB 3000	<242>	30460 RETURN	<034>	33060 N=(M AND 128)/128	<189>
28220 X=PEEK(O+Y)	<018>	30470 :	<240>	33070 RETURN	<107>
28230 N=(X AND 128)/128	<026>	30500 REM *** CMP ABSOLUTE	<180>	33190 :	<145>
28240 Z--(X=0)	<218>	30510 GOSUB 4000	<008>	33200 REM *** INX	<250>
28250 RETURN	<112>	30520 M=PEEK(OO)	<196>	33210 X=(X+1) AND 255	<206>
28390 :	<170>	30530 GOSUB 8400	<124>	33220 Z--(X=0)	<117>
28400 REM *** CLV	<142>	30540 RETURN	<116>	33230 N=(X AND 128)/128	<201>
28410 V=0	<199>	30590 :	<084>	33240 RETURN	<021>
28420 RETURN	<028>	30600 REM *** DEC ABSOLUTE	<072>	33290 :	<247>
28490 :	<016>	30610 GOSUB 4000	<108>	33300 REM *** SBC #IMMEDIATE	<203>
28500 REM *** LDA ABSOLUTE,Y	<065>	30620 M=PEEK(OO)	<040>	33310 GOSUB 3000	<007>
28510 GOSUB 4000	<040>	30630 M=(M-1) AND 255	<087>	33320 M=0	<235>
28520 A=PEEK(OO+Y)	<139>	30640 POKE OO,M	<066>	33330 GOSUB 8200	<099>
28530 N=(A AND 128)/128	<101>	30650 Z--(M=0)	<147>	33340 RETURN	<123>
28540 Z--(A=0)	<066>	30660 N=(M AND 128)/128	<072>	33390 :	<091>
28550 RETURN	<158>	30670 RETURN	<246>	33400 REM *** NOP	<125>
28590 :	<116>	30790 :	<030>	33460 RETURN	<243>
28600 REM *** TSX	<034>	30800 REM *** BNE	<178>	33590 :	<037>
28610 X=S	<203>	30810 GOSUB 3000	<046>	33600 REM *** CPX ABSOLUTE	<115>
28620 Z--(X=0)	<088>	30820 IF Z=0 THEN GOSUB 7000	<012>	33610 GOSUB 4000	<061>
28630 N=(X AND 128)/128	<172>	30830 RETURN	<152>	33620 M=PEEK(OO)	<249>
28640 RETURN	<248>	30890 :	<130>	33630 C--((X-M)>=0)	<237>
28790 :	<062>	30900 REM *** CMP (INDIRECT),Y	<239>	33640 M=(X-M) AND 128/128	<080>
28800 REM *** LDY ABSOLUTE,X	<131>	30910 GOSUB 3000	<146>	33650 Z--((X-M)>=0)	<093>
28810 GOSUB 4000	<086>	30920 OO=PEEK(O)+256*PEEK(O+1)	<212>	33660 RETURN	<189>
28820 Y=PEEK(OO+X)	<023>	30930 M=PEEK(OO)	<096>	33690 :	<137>
28830 N=(Y AND 128)/128	<150>	30940 GOSUB 8400	<024>	33700 REM *** SBC ABSOLUTE	<191>
28840 Z--(Y=0)	<118>	30950 RETURN	<016>	33710 GOSUB 4000	<161>
28850 RETURN	<204>	31290 :	<022>	33720 M=PEEK(OO)	<093>
28890 :	<162>	31300 REM *** CMP ZEROPAGE,X	<190>	33730 GOSUB 8200	<245>
28900 REM *** LDA ABSOLUTE,X	<207>	31310 GOSUB 3000	<038>	33740 RETURN	<013>
28910 GOSUB 4000	<186>	31320 M=PEEK(O+X)	<025>	33790 :	<237>
28920 A=PEEK(OO+X)	<027>	31330 GOSUB 8400	<162>	33800 REM *** INC ABSOLUTE	<169>
28930 N=(A AND 128)/128	<249>	31340 RETURN	<154>	33810 GOSUB 4000	<007>
28940 Z--(A=0)	<214>	31390 :	<122>	33820 M=PEEK(OO)	<195>
28950 RETURN	<050>	31400 REM *** DEC ZEROPAGE,X	<082>	33830 M=(M+1) AND 255	<178>
28990 :	<008>	31410 GOSUB 3000	<138>	33840 POKE OO,M	<221>
29000 REM *** LDX ABSOLUTE,Y	<080>	31420 M=PEEK(O+X)	<125>	33850 Z--(M=0)	<046>
29010 GOSUB 4000	<032>	31430 M=(M-1) AND 255	<125>	33860 N=(M AND 128)/128	<227>
29020 X=PEEK(OO+Y)	<223>	31440 POKE OO,X,M	<002>	33870 RETURN	<145>
29030 N=(X AND 128)/128	<064>	31450 Z--(M=0)	<185>	33990 :	<183>
29040 Z--(X=0)	<000>	31460 N=(M AND 128)/128	<110>	34000 REM *** BEQ	<210>
29050 RETURN	<150>	31470 RETURN	<028>	34010 GOSUB 3000	<199>
29190 :	<210>	31590 :	<068>	34020 IF Z=1 THEN GOSUB 7000	<197>
29200 REM *** CPY #IMMEDIATE	<191>	31600 REM *** CLD	<022>	34030 RETURN	<049>
29210 GOSUB 3000	<226>	31610 D=0	<025>	34090 :	<029>
29220 C--((Y-O)>=0)	<020>	31620 RETURN	<180>	34100 REM *** SBC (INDIRECT),Y	<252>
29230 N=((Y-O) AND 128)/128	<057>	31690 :	<168>	34110 GOSUB 3000	<045>
29240 Z--((Y-O)=0)	<057>	31700 REM *** CMP ABSOLUTE,Y	<041>	34120 OO=PEEK(O)+256*PEEK(O+1)	<111>
29250 RETURN	<096>	31710 GOSUB 4000	<192>	34130 M=PEEK(OO)	<251>
29290 :	<054>	31720 M=PEEK(OO+Y)	<083>	34140 GOSUB 8200	<147>
29300 REM *** CMP (INDIRECT,X)	<013>	31730 GOSUB 8400	<052>	34150 RETURN	<171>
29310 GOSUB 3000	<070>	31740 RETURN	<044>	34490 :	<175>
29320 OO=PEEK(O+X)+256*PEEK(O+X+1)	<180>	32090 :	<060>	34500 REM *** SBC ZEROPAGE,X	<201>
29330 M=PEEK(OO)	<020>	32100 REM *** CMP ABSOLUTE,X	<185>	34510 GOSUB 3000	<191>
29340 GOSUB 8400	<204>	32110 GOSUB 4000	<084>	34520 M=PEEK(O+X)	<178>
29350 RETURN	<196>	32120 M=PEEK(OO+X)	<229>	34530 GOSUB 8200	<027>
29590 :	<100>	32130 GOSUB 8400	<200>	34540 RETURN	<051>
29600 REM *** CPY ZEROPAGE	<222>	32140 RETURN	<192>	34590 :	<021>
29610 GOSUB 3000	<116>	32190 :	<160>	34600 REM *** INC ZEROPAGE,X	<179>
29620 M=PEEK(O)	<252>	32200 REM *** DEC ABSOLUTE,X	<077>	34610 GOSUB 3000	<037>
29630 C--((Y-M)>=0)	<172>	32210 GOSUB 4000	<184>	34620 M=PEEK(O+X)	<024>
29640 N=((Y-M) AND 128)/128	<211>	32220 M=PEEK(OO+X)	<073>	34630 M=(M+1) AND 255	<216>
29650 Z--((Y-M)=0)	<209>	32230 M=(M-1) AND 255	<163>	34640 POKE OO,X,M	<157>
29690 :	<200>	32240 POKE OO,X,M	<132>	34650 Z--(M=0)	<084>
29700 REM *** CMP ZEROPAGE	<185>	32250 Z--(M=0)	<223>	34660 N=(M AND 128)/128	<009>
29710 GOSUB 3000	<218>	32260 N=(M AND 128)/128	<150>	34670 RETURN	<183>
29720 M=PEEK(O)	<098>	32270 RETURN	<068>	34790 :	<221>
29730 GOSUB 8400	<086>	32390 :	<106>	34800 REM *** SED	<047>
29740 RETURN	<078>	32400 REM *** CPX #IMMEDIATE	<086>	34810 D=1	<194>
29790 :	<046>	32410 GOSUB 3000	<122>	34820 RETURN	<079>
29800 REM *** DEC ZEROPAGE	<077>	32420 C--((X-O)>=0)	<044>	34890 :	<067>
29810 GOSUB 3000	<062>	32430 N=((X-O) AND 128)/128	<145>	34900 REM *** SBC ABSOLUTE,Y	<054>
29820 M=PEEK(O)	<198>	32440 Z--((X-O)=0)	<081>	34910 GOSUB 4000	<091>
29830 M=(M-1) AND 255	<049>	32450 RETURN	<248>	34920 M=PEEK(OO+Y)	<238>
29840 POKE O,M	<187>	32490 :	<206>	34930 GOSUB 8200	<175>
29850 Z--(M=0)	<109>	32500 REM *** SBC (INDIRECT,X)	<023>	34940 RETURN	<199>
29860 N=(M AND 128)/128	<034>	32510 GOSUB 3000	<222>	35290 :	<213>
29870 RETURN	<208>	32520 OO=PEEK(O+X)+256*PEEK(O+X+1)	<078>	35300 REM *** SBC ABSOLUTE,X	<196>
29990 :	<248>	32530 M=PEEK(OO)	<174>	35310 GOSUB 4000	<237>
30000 REM *** INY	<098>	32540 GOSUB 8200	<070>	35320 M=PEEK(OO+X)	<126>
30010 Y=(Y+1) AND 255	<089>	32550 RETURN	<094>	35330 GOSUB 8200	<067>
30020 Z--(Y=0)	<028>	32790 :	<255>	35340 RETURN	<091>
30030 N=(Y AND 128)/128	<080>	32800 REM *** CPX ZEROPAGE	<041>	35390 :	<059>
30040 RETURN	<124>	32810 GOSUB 3000	<015>	35400 REM *** INC ABSOLUTE,X	<174>
30090 :	<092>	32820 M=PEEK(O)	<151>	35410 GOSUB 4000	<083>
30100 REM *** CMP #IMMEDIATE	<190>	32830 C--((X-M)>=0)	<199>	35420 M=PEEK(OO+X)	<228>
30110 GOSUB 3000	<108>	32840 N=((C-M) AND 128)/128	<232>	35430 M=(M+1) AND 255	<254>
30120 M=0	<080>	32850 Z--((X-M)=0)	<236>	35440 POKE OO,X,M	<031>
30130 GOSUB 8400	<232>	32860 RETURN	<151>	35450 Z--(M=0)	<122>
30140 RETURN	<224>	32890 :	<099>	35460 N=(M AND 128)/128	<047>
30190 :	<192>	32900 REM *** SBC ZEROPAGE	<196>	35470 RETURN	<221>
30200 REM *** DEX	<099>	32910 GOSUB 3000	<115>	35500 :	<169>
30210 X=(X-1) AND 255	<063>	32920 M=PEEK(O)	<251>	49000 FOR HH=34768 TO 34783:RE	
30220 Z--(X=0)	<166>	32930 GOSUB 8200	<207>	AD H:POKE HH,H:NEXT:RETU	
30230 N=(X AND 128)/128	<250>	32940 RETURN	<231>	RN	<055>
30240 RETURN	<147>	32990 :	<199>	50000 DATA 9,240,8,8,88,8,8,8,	
30390 :	<138>	33000 REM *** INC ZEROPAGE	<172>	8,8,8,8,8,0,0,0	<123>
30400 REM *** CPY ABSOLUTE	<217>	33010 GOSUB 3000	<215>		
30410 GOSUB 4000	<162>	33020 M=PEEK(O)	<095>		
30420 M=PEEK(OO)	<094>	33030 M=(M+1) AND 255	<140>		
30430 C--((Y-M)>=0)	<210>	33040 POKE O,M	<086>		
30440 N=((Y-M) AND 128)/128	<249>				

READY.

Listing »Demo 6502« (Schluß)