

Herwig Feichtinger

65XX-Befehls-satz

Die Prozessoren der Familie 65XX haben alle den gleichen Befehlssatz, der hier übersichtlich zusammengestellt ist. Zu dieser CPU-Familie gehören u. a. die Typen 6502 (AIM-65, PC-100, KIM-1, CBM, PET, Apple-II/III, Pearcom u. a. Computer) und 6504 (im Einplatinencomputer EMUF).

Bei den Befehlen mit mehreren Adressierungsarten ist die Beeinflussung des Statusregisters mit angegeben; bei den übrigen Befehlen wurde sie der Übersichtlichkeit halber weggelassen (PLA wie LDA, PHA wie STA, DEX wie DEC usw.). Die Interrupt-Vektoren haben folgende festverdrahtete Adressen:

- FFFA NMI Low (nicht bei 6504)
- FFFB NMI High (nicht bei 6504)
- FFFC RES Low
- FFFD RES High
- FFFE IRQ Low
- FFFF IRQ High



Register des VIA-Bausteins 6522

Adresse	Register
XXX0	Datenregister Port B
XXX1	Datenregister Port A, mit Handshake
XXX2	Port-B-Datenrichtungsregister
XXX3	Port-A-Datenrichtungsregister
XXX4	Timer 1 Low lesen/schreiben
XXX5	Timer 1 High lesen/schreiben/starten
XXX6	Timer-1-Latch Low
XXX7	Timer-1-Latch High
XXX8	Timer 2 Low lesen/schreiben IRQ rücksetzen
XXX9	Timer 2 High lesen/schreiben IRQ rücksetzen
XXXA	Schieberegister
XXXB	ACR (Steuer-Hilfsregister) lesen
XXXC	PCR (Peripherie-Steuerregister) lesen
XXXD	IFR (Interrupt-Flagregister) lesen
XXXE	IER (Interrupt-Enable-Register) lesen
XXXF	Port-A-Datenregister (ohne Handshake)

Register des RIOT-Bausteins 6532

Adresse	Register
XXX0	Datenregister Port A
XXX1	Datenrichtungsregister Port A
XXX2	Datenregister Port B
XXX3	Datenrichtungsregister Port B
XXX6	Timer lesen
XXX7	Interrupt-Flags lesen/löschen
XX14-17	Timer schreiben, 1/8/64/1024
XX1C-1F	Wie 14-17, jedoch mit Interrupt
XX04	PA-7-Interrupt bei H-L-Flanke programmieren
XX05	PA-7-Interrupt bei L-H-Flanke programmieren
XX06	PA-7-Interrupt sperren

Befehle m. mehreren Adressierungsarten

Mn.	Im	Ab	ZP	Ac	X)	Y)	ZX	ZY	AX	AY	NZCV
ADC	69	6D	65	--	61	71	75	--	7D	79	xxxx
AND	29	2D	25	--	21	31	35	--	3D	39	xx--
ASL	--	0E	06	0A	--	--	16	--	1E	--	xxx-
BIT	--	2C	24	--	--	--	--	--	--	--	xx-x
CMP	C9	CD	C5	--	C1	D1	D5	--	DD	D9	xxx-
CPX	E8	EC	E4	--	--	--	--	--	--	--	xxx-
CPY	C8	CC	C4	--	--	--	--	--	--	--	xxx-
DEC	--	CE	C6	--	--	--	D6	--	DE	--	xx--
EOR	49	4D	45	--	41	51	55	--	5D	59	xx--
INC	--	EE	E6	--	--	F6	--	--	FE	--	xx--
LDA	A9	AD	A5	--	A1	B1	B5	--	BD	B9	xx--
LDX	A2	AE	A6	--	--	--	B6	--	BE	--	xx--
LDY	A0	AC	A4	--	--	--	B4	--	BC	--	xx--
LSR	--	4E	46	4A	--	--	56	--	5E	--	xxx-
ORA	09	0D	05	--	01	11	15	--	1D	19	xx--
ROL	--	2E	26	2A	--	--	36	--	3E	--	xxx-
ROR	--	6E	66	6A	--	--	76	--	7E	--	xxx-
SBC	E9	ED	E5	--	E1	F1	F5	--	FD	F9	xxxx
STA	--	8D	85	--	81	91	95	--	9D	99	----
STX	--	8E	86	--	--	--	96	--	--	--	----
STY	--	8C	84	--	--	--	94	--	--	--	----

Befehle mit nur einer Adressierungsart

BCC	90	BCS	98	BEQ	F0	BMI	30
BNE	D0	BPL	10	BRK	00	BVC	50
BVS	70	CLC	18	CLD	D8	CLI	58
CLV	B8	DEX	CA	DEY	88	INX	E8
INY	C8	JMI	6C	JMP	4C	JSR	20
NOP	EA	PHA	48	PHP	08	PLA	68
PLP	28	RTI	40	RTS	60	SEC	38
SED	F8	SEI	78	TAX	AA	TAY	AB
TSX	BA	TXA	8A	TXS	9A	TYA	9B

Adressierungsarten
 Im=Immediate; Ab=Absolute; ZP=ZeroPage;
 Ac=Accu; X)=x-indiziert indirekt; Y)=
 indirekt y-indiziert; ZX=ZeroPage x-
 indiziert; ZY=ZeroPage y-indiziert;
 AX=absolut x-indiziert; AY=absolut y-
 indiziert.

Flags im Statusregister
 N=Negativ (Bit 7); Z=Zero; C=Carry; I=
 Interrupt-Disable; D=Decimal Mode; V=
 Overflow (Bit 6).

Tabelle: So sind die Anschlüsse des User-Ports belegt. PIA 1 ist der Interface-Baustein ab Adresse \$E810, VIA der ab \$E840

Funktion	Anschluß Rechner	intern
Reset	J2-5 PA 7	PIA 1
Clock	J2-4 CA 2	PIA 1
CS (PD/PGM)	J2-7 PB 3	VIA
PD/PGM (A 11)	J2-M CB 2	VIA
Data 7	J2-L PA 7	VIA
Data 6	J2-K PA 6	VIA
Data 5	J2-J PA 5	VIA
Data 4	J2-H PA 4	VIA
Data 3	J2-F PA 3	VIA
Data 2	J2-E PA 2	VIA
Data 1	J2-D PA 1	VIA
Data 0	J2-C PA 0	VIA

schnell Verwendungsmöglichkeiten für die vielen freien Bytes innerhalb der 4000er ROMs finden. Schaut man sich dieses Betriebssystem nämlich einmal an, dann wird man feststellen, daß über 1,5 K der 18 K mit dem hexadezimalen Wert \$AA programmiert und demzufolge unbenutzt sind. Verwendet man statt der Commodore-ROMs nun eigene EPROMs, so lassen sich in dem freien Raum eine Vielzahl nützlicher Routinen oder kleinerer Spracherweiterungen unterbringen, ohne daß man dafür in den wertvollen Bereich von \$9000...\$AFF zu gehen braucht. Leider ist Commodore bei den neueren 4000er-Geräten dazu übergegangen, die ROMs nicht mehr zu sockeln, wohl aus Kostengründen. Und es ist nicht unbedingt zu empfehlen, hier mit dem Lötcolben Abhilfe schaffen zu wollen. Schade, denn durch diese Maßnahme verliert der CBM 4000 eine gehörige Portion an Flexibilität, die doch eigentlich keinem Gerät schaden kann. Wohl dem, der noch einen der alten 3000er besitzt...

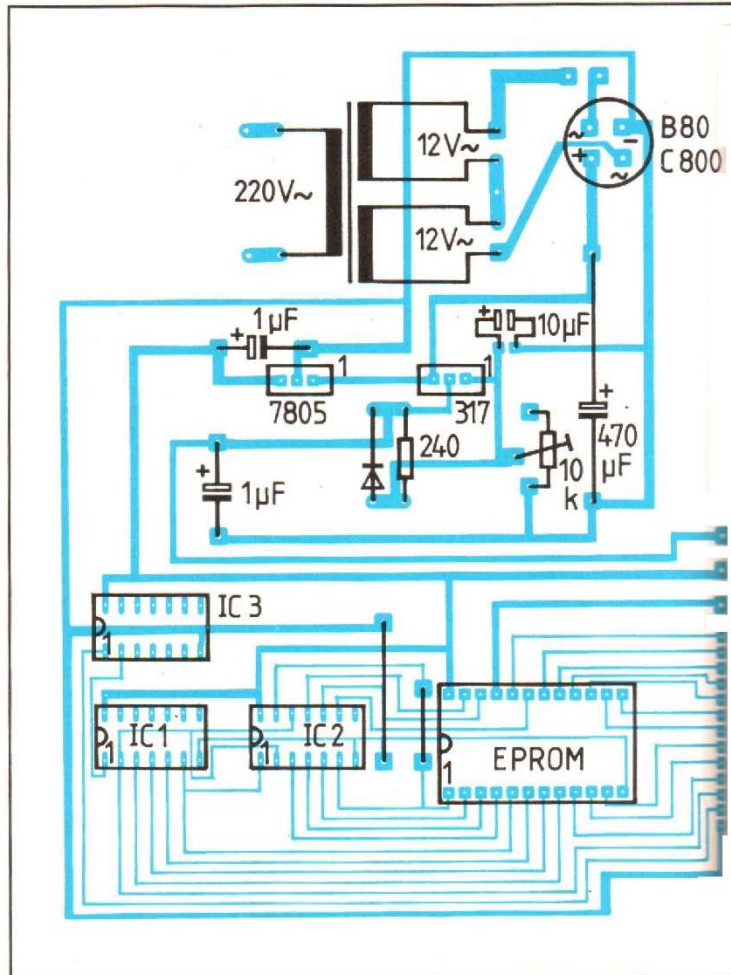


Bild 6. Der Bestückungsplan des Programmers. Für die EPROMs wird sinnvollerweise Wechselsockel eingesetzt

AIM-65/PC-100: Zusammenarbeit Forth und Editor

Das folgende Programm in Forth erleichtert die Arbeit mit dem Editor bei der Bearbeitung von Forth-Programmen. Nach dem Initialisieren des Editors und Laden des Programms ins Forth-Dictionary muß nicht mehr zwischen Editor,

Monitor und Forth hin- und hergesprungen werden. Durch den Editorbefehl L IN=U ist man sofort in Forth und kann

den Text im Editor durch S IN=M Die Rückkehr in den Editor erfolgt das Forth-Wort E. Thomas

Das ist das Forthprogramm „Zusammenarbeit“

```

= <L>          ( EDITOR T )          = <L>
/              CODE E                /
OUT=P          F6CF JMP,             OUT=U
FORGET TASK   END-CODE               AIM 65 FORTH
HEX           ( NEW VALUE FENCE )    S IN=M
B003 010A !   HERE FENCE !          OK
: S           TASK ;                 E
R) R) R)     FINIS                    FORGET TASK
00E3 @ 00DF !
SOURCE ;
    
```