

Programmieren von EPROMs

Dipl.-Ing. GERHARD NICKLISCH
und Dr.-Ing. WOLFGANG WINTER

Mitteilung aus der Sektion Informationstechnik
der TU Dresden

In diesem Beitrag wird ein mit geringem Aufwand zu realisierendes, mikrorechnergesteuertes Programmiergerät für EPROMs der Typenreihe 2708 (8708) vorgestellt. Dabei werden neben dem Hardware-Aufbau die zur Programmierung erforderlichen Algorithmen und ihre programmtechnische Umsetzung erläutert.

Programmierschritt

Die Datenausgänge des unprogrammierten EPROMs 2708 (s. Bilder 1 und 2 und Tafel 1) befinden sich im Zustand 1 ($\cong H$). Beim Programmieren muß somit log. 0 auf die gewünschten Positionen geschrieben werden. Der dafür erforderliche Ablauf ist im Bild 3 dargestellt. Er ist durch die N-malige, jeweils aufeinanderfolgende Programmierung aller 1024 Speicherplätze (Byte-Format) des EPROM gekennzeichnet. Die exakten zeitlichen Bedingungen sind dem Bild 4 und der Tafel 2 zu entnehmen.

Während der Adressierung jedes Speicherplatzes wird die am EPROM-Eingang „Program“ liegende Spannung U_p für die Zeit t_{pw} auf den Wert U_{IHP} ($25V \leq U_{IHP} \leq 27V$) angehoben. Außerdem liegt während des gesamten Programmiervorgangs am Eingang $\overline{CS}/\overline{WE}$ die Spannung $U_{CS/WE} = U_{IHW}$ ($11,4V \leq U_{IHW} \leq 12,6V$).

Der einmalige Durchlauf aller Speicherplätze wird als Programmierschleife bezeichnet. Die Anzahl der erforderlichen Programmierschleifen N hängt von der Programmierimpulsdauer t_{pw} ab. Dabei wird gefordert:

$$N t_{pw} = 100 \text{ ms} \quad (1)$$

$$0,1 \text{ ms} \leq t_{pw} \leq 1 \text{ ms}$$

Für die vollständige Programmierung eines 2708 ist somit mindestens die Zeit T_{Pmin} erforderlich:

$$T_{Pmin} = 1024 \cdot 100 \text{ ms} = 102,4 \text{ s} \quad (2)$$

Soll der Speicher unmittelbar nach erfolgter Programmierung gelesen werden, so ist insbesondere auf die Einhaltung der folgenden zwei Bedingungen zu achten:

- Der Übergang $U_{CS/WE} = U_{IHW} \rightarrow U_{CS/WE} = U_{IL}$ muß nach dem Ende des letzten Programmierimpulses erfolgen.
- Der zu lesende Speicherplatz darf erst nach dem Übergang $U_{CS/WE} = U_{IHW} \rightarrow U_{CS/WE} = U_{IL}$ adressiert werden.

Werden diese Bedingungen nicht beachtet, so liegen auf Grund von Ladungsansammlungen am Schaltkreisausgang für die Dauer mehrerer Millisekunden ungültige Daten (Verlängerung von t_{DPR}).

Programmiergerät

Das vorgestellte Programmiergerät läßt sich direkt in 8080- oder 8085-Mikrorechnersysteme integrieren. Da die Spannungen U_{CC} , U_{BB} und U_{DD} bei Verwendung von 2708-Bausteinen bzw. des CPU-Schaltkreises 8080 bereits im System existieren, muß nur die Programmierspannung von +26V zusätzlich erzeugt werden. Das kann z. B. durch einen Transverter geschehen. Die Schaltung des Programmiergerätes ist in den Bildern 5 und 6 dargestellt [4]. Der erforderliche Hardware-Aufwand konnte gegenüber anderen Varianten [3] durch Verwendung des programmierbaren Peri-

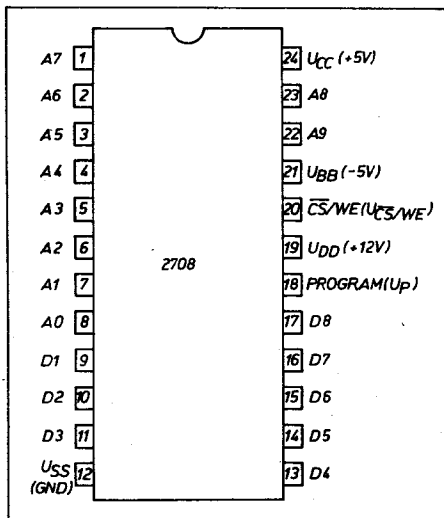


Bild 1: Pin-Belegung des EPROM 2708

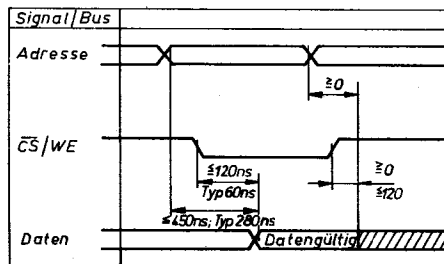
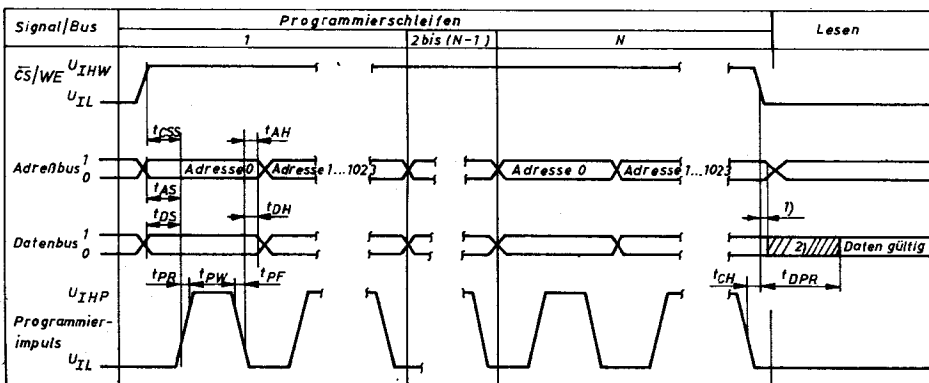


Bild 2: 2708-Lesezyklus (Zeitangaben in ns)

Bild 4: Signalfolge bei der 2708-Programmierung (Zeitangaben in Tafel 3)

- 1) Der Übergang von $U_{CS/WE} = U_{IHW} \rightarrow U_{CS/WE} = U_{IL}$ muß nach dem Ende des Programmierimpulses und vor einer Adreßänderung erfolgen
- 2) Daten ungültig ∇



pheriebausteins 8255/8255 A stark reduziert werden.

Port A sowie die beiden niederwertigen Positionen von Port C (PC1, PC0) des 8255 werden für das Aussenden der EPROM-Adresse benutzt, während der Datentransport in beiden Richtungen über das Port B erfolgt. Die jeweilige Übertragungsrichtung wird durch entsprechende Änderungen des 8255-Steuerwortes eingestellt.

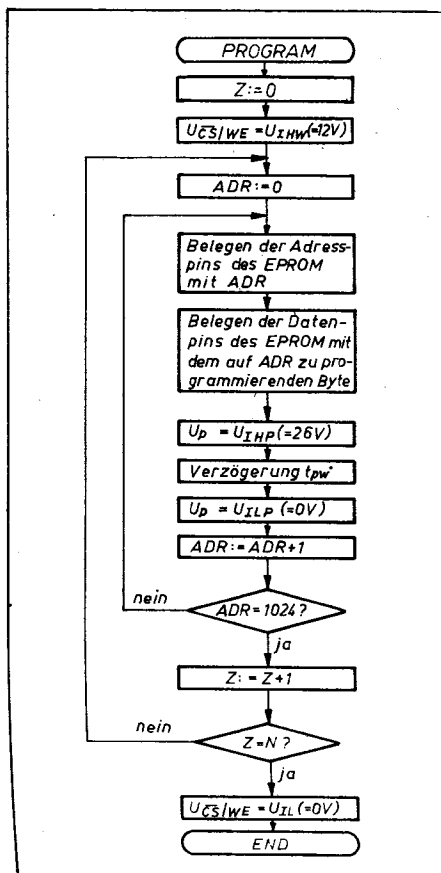


Bild 3: Programmialgorithmus (s. a. Bild 4, Tafeln 2 und 3)

ADR – EPROM-Adresse (Deximaldarstellung); Z – Zähler der Programmierschleifen; N – Anzahl der Programmierschleifen; U_p – Spannung am EPROM-Eingang „Program“; $U_{CS/WE}$ – Spannung am EPROM-Eingang $\overline{CS}/\overline{WE}$; t_{pw} – Dauer des Programmierimpulses

Da der Eingangs-H-Pegel U_{IH} des EPROM im Intervall $3V \leq U_{IH} \leq U_{CC} + 1V$ liegt, müssen alle Leitungen, die vom 8255 zum Speicher führen, über Pull-up-Widerstände ($3,3 \text{ k}\Omega$) mit U_{CC} verbunden werden. Die Positionen PC7 bis PC4 von Port C dienen zur Steuerung der Spannungen U_p und $U_{CS/WE}$ (s. Bild 6). Dabei gilt:

$$U_p = \begin{cases} 26 \text{ V, wenn } \overline{PC5} \wedge PC4 = 1 \\ 0 \text{ V sonst} \end{cases} \quad (3)$$

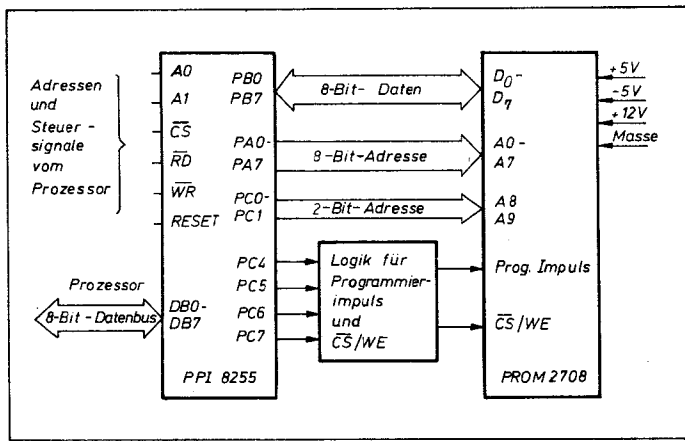


Bild 5: Prinzipdarstellung des Programmiergerätes

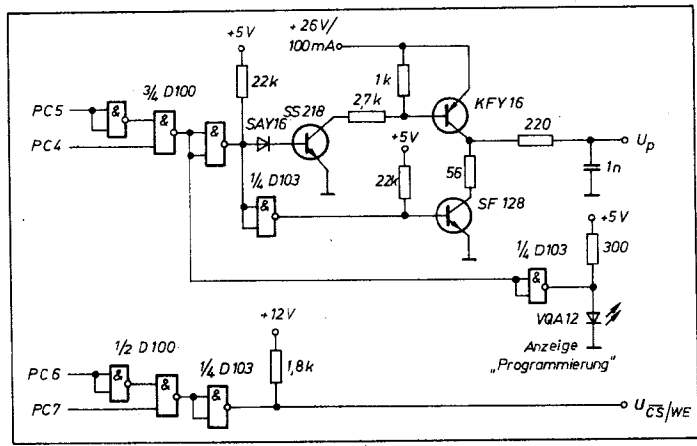


Bild 6: Logik für Programmierimpuls und CS/WE

$$U_{CS/WE} = \begin{cases} 12 \text{ V, wenn } PC7 \wedge \overline{PC6} = 1 \\ 0 \text{ V sonst} \end{cases} \quad (4)$$

Es wurden Bitkombinationen zur Steuerung benutzt, um die Spannungen eindeutig durch Software festlegen zu können. Eine Steuerung durch Einzelbits ist nicht möglich, da

- nach dem Rücksetzen des 8255 alle Portleitungen hochohmig sind (Eingaberichtung)
- nach dem Festlegen der Betriebsart (Mode) alle Port-Ausgänge L-Pegel führen.

In der Tafel 3 sind die Funktionen der 8255-Ports zusammenfassend dargestellt.

Das Programmiergerät wurde als Einschub für das im Bild 7 dargestellte 8080-Mehrrechnersystem realisiert. Auf der Frontseite des Einschubes befinden sich die Klemmfassung für den zu programmierenden EPROM sowie zwei Leuchtdioden zur Anzeige des Programmierzustandes ($U_P = 26 \text{ V}$) und der Programmierspannung von 26 V .

Programmier-Software

Es werden zwei Programmiervarianten PROMPRO 1 und PROMPRO 2 beschrieben. Ihr prinzipieller Aufbau ist in den Bildern 8 und 9 dargestellt.

Im Programm PROMPRO 1 wird exakt die von den Speicherherstellern angegebene Programmiervorschrift eingehalten. Es wurden die Werte $N = 200$ und $t_{PW} = 500 \mu\text{s}$ gewählt, so daß $N \cdot t_{PW} = 100 \text{ ms}$ gilt.

Im Anschluß an den Programmieralgorithmus wird ein Soll-Istwert-Vergleich vorgenommen. Sollten sich dabei Abweichungen ergeben, so werden diese durch ein Fehlerausgabeprogramm zur Anzeige gebracht. Wie eigene Versuche und Literaturangaben [4] zeigten, ist es nicht grundsätzlich erforderlich, eine integrale Programmierzeit von 100 ms /Speicherplatz einzuhalten. Das experimentelle Programm PROMPRO 2 berücksichtigt diese Tatsache. Es ermöglicht 16 Programmierversuche PV, wobei für jeden Versuch gilt:

$$N = 12, t_{PW} = 150 \mu\text{s}, N \cdot t_{PW} = 1,8 \text{ ms}$$

Nach jedem Versuch wird getestet, ob die Programmierung des EPROM erfolgreich war (Soll-Istwert-Vergleich). Ist dies nicht der Fall und wurden noch nicht alle 16 Programmierversuche ausgeführt, so erfolgt eine Wiederholung des Programmieralgorithmus.

Bei positivem Testergebnis wird das Programm dagegen nach einem abschließen-

Tafel 1: Pin-Funktionen des 2708 bei den Betriebsarten Lesen, Programmieren und Deselect

$$11,4 \text{ V} \leq U_{IHV} \leq 12,6 \text{ V}; 3,0 \text{ V} \leq U_{IH} \leq U_{CC} + 1 \text{ V}; U_{SS} \leq U_{IL} \leq 0,65 \text{ V}; 25 \text{ V} \leq U_{IHP} \leq 27 \text{ V}; U_{NS} \leq U_{ILP} \leq 1 \text{ V}$$

Pin-Bezeichnung	D0 bis D7	A0 bis A9	Programm	\overline{CS} , WE	U_{SS}	U_{RB}	U_{CC}	U_{DD}
Lesen	Datenausgang	Adresseneingang	U_{IL}	U_{IL}	GND	-5V	+5V	+12V
Programmieren	Dateneingang	Adresseneingang	Programmierimpuls	U_{IHW}	GND	-5V	+5V	+12V
Deselect	hochohmig	Don't care	U_{IH}	U_{IH}	GND	-5V	+5V	+12V

Tafel 3: Funktionen der drei 8255-Ports

Pi7 bis Pi0 = Bus von Port i (i = A, B, C); A9 bis A0 = EPROM-Adreßbus; D7 bis D0 = EPROM-Datenbus; \emptyset = Don't care

Port i	Mode	Richtung	Steuerwort	Pi7	Pi6	Pi5	Pi4	Pi3	Pi2	Pi1	Pi0	Bemerkung
Lesen	A	0	Ausgabe	A7	A6	A5	A4	A3	A2	A1	A0	$U_P = 0 \text{ V}; U_{CS/WE} = 0 \text{ V}$
	B	0	Eingabe	D7	D6	D5	D4	D3	D2	D1	D0	
	C	-	Ausgabe	0	0	0	0	\emptyset	\emptyset	A9	A8	
Programmieren	A	0	Ausgabe	A7	A6	A5	A4	A3	A2	A1	A0	$U_P = 0 \text{ V}; U_{CS/WE} = 12 \text{ V}; U_P = 26 \text{ V}; U_{CS/WE} = 12 \text{ V}$
	B	0	Ausgabe	D7	D6	D5	D4	D3	D2	D1	D0	
	C	-	Ausgabe	1	0	1	0	\emptyset	\emptyset	A9	A8	
	C	-	Ausgabe	1	0	0	1	\emptyset	\emptyset	A9	A8	

den, aus Sicherheitsgründen vorgesehenen Programmierzyklus beendet. Sollte der Speicher auch nach 16 Programmierversuchen noch nicht die gewünschten Daten enthalten, wird das Programm nach einer Fehlerausgabe abgebrochen.

Nachfolgend werden die beiden Programmvarianten näher erläutert, wobei auf den bereits vorgestellten Hardware-Aufbau des Programmiergerätes Bezug genommen wird. Die Programmbeschreibung erfolgt durch Struktogramme [1] [2]. Dabei werden weitgehend die Eigenarten des zur Programmierung benutzten 8080-Mikrorechners berücksichtigt. Insbesondere gelten:

- A - 8080-Akkumulator (8 bit)
- B, C - 8080-Register (8 bit)
- DE, HL - 8080-Registerpaare ($2 \times 8 \text{ bit}$)
- <HL> - Inhalt des durch das Registerpaar HL adressierten Speicherplatzes (indirekte Adressierung)
- IN(Port i) - Laden des Akkumulators mit dem an Port i des 8255 liegenden Datenwort

OUT(Port i) - Ausgabe des Akkumulatorinhaltes an Port i des 8255

CALL, UPx - Aufruf des Unterprogramms UPx

RETURN - Rückkehr ins Hauptprogramm
Z - Zero-Flag

Weiterhin sei vereinbart:

- Die zu programmierenden Daten befinden sich im RAM-Speicherbereich (Quellspeicher) mit der Anfangsadresse XXXXH.

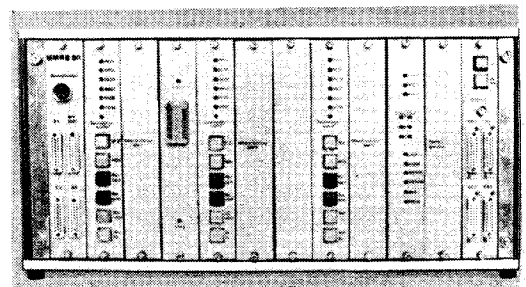


Bild 7: Einsatz des Programmiergerätes in einem 8080-Dreirechnersystem

- Die Adressierung des Quellspeichers erfolgt durch das Registerpaar HL (indirekte Adressierung).
- Die jeweilige EPROM-Adresse befindet sich im Registerpaar DE.
- Das Register B dient als Zähler der Programmierschleifen (N).
- Das Register C dient als Zähler der Programmierversuche (PV).

Die Programme setzen sich aus Haupt- und Unterprogrammen zusammen. Das Hauptprogramm PROMPRO 1 ist im Bild 10 dargestellt. Es ruft nacheinander die Unterprogramme PROGRAM (Bild 11) und FEHLER (Bild 12) auf.

Zu Beginn der Unterprogramme wird entsprechend Tafel 3 die jeweilige Betriebsart des 8255 eingestellt. Die Adresse des EPROM wird über Port A und Port C (Positionen PC1, PC0) bereitgestellt, während Port B entweder zur Datenausgabe oder zum Lesen des Speicherinhaltes dient.

Die Einstellung der Spannungen U_P und $U_{CS/WE}$ erfolgt über die Positionen PC7 bis PC4 von Port C. Dabei gilt die in der Tafel 4 gewählte Zuordnung (s. a. Bilder 5, 6 und Tafel 3).

Die Dauer t_{PW} des Programmierimpulses wird im Unterprogramm PROGRAM durch eine Zeitschleife (ZS1) bestimmt. Sie hängt somit von der Taktperiode T des benutzten Rechners ab. Der hier gewählte Startwert $A=35H$ für $t_{PW}=500\mu s$ bzw. $A=10H$ für $t_{PW}=150\mu s$ gilt unter der Bedingung $T \approx 500 ns$.

Das Unterprogramm FEHLER vergleicht den Inhalt des EPROM mit den im Quellspeicherbereich befindlichen zu programmierenden Daten. Bei festgestellten Abweichungen werden jeweils die Adresse des fehlerhaft programmierten Speicherplatzes sowie die Soll- und Istdaten ausgegeben (Adresse in DE, Istdaten in A, Solldaten in HL).

Bild 13 zeigt den Ablauf des Hauptprogramms PROMPRO 2. Es ruft neben den Programmen PROGRAM und FEHLER das zusätzliche Unterprogramm COMPARE (Bild 14) auf. Dieses testet, ob der Speicher bereits vollständig programmiert ist.

Die praktisch ermittelte Programmierzeit beträgt bei Verwendung der Programme PROMPRO 1 und PROMPRO 2 103 s bzw. 5 s (nur ein PV erforderlich). Speicher, die mit Hilfe des Programms PROMPRO 2 pro-

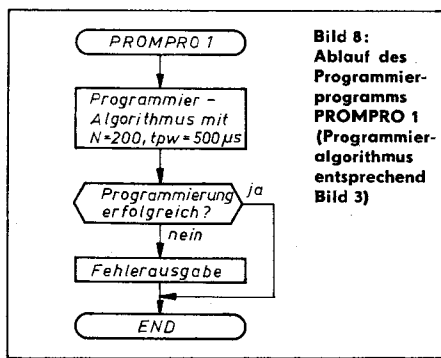


Bild 8: Ablauf des Programmierprogramms PROMPRO 1 (Programmieralgorithmus entsprechend Bild 3)

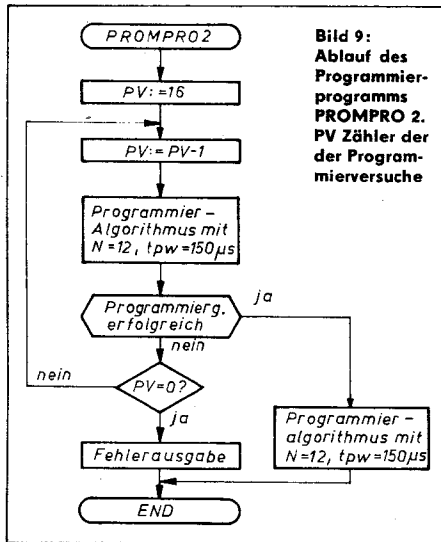


Bild 9: Ablauf des Programmierprogramms PROMPRO 2. PV Zähler der Programmierversuche

```
PROMPRO 1
/* EPROM-Programmierung
CALL, PROGRAM
/* N = 200, tpw = 500µs
CALL, FEHLER
/* Soll-/Istwertvergleich mit Fehlerausgabe
END
```

Bild 10: Hauptprogramm PROMPRO 1

```
PROGRAM
/* Programmieralgorithmus
A := 80H
OUT (Steuerlogik 8255)
/* Port B in Ausgaberrichtung
A := A0H
OUT (Port C)
/* UCS/WE = 12V, Up = 0V
B := CBH (0CH)
/* N = 200 (12)
HL := XXXXH - 1
DE := FFFFH
HL := HL + 1
DE := DE + 1
A := D
A := A v A0H
OUT (Port C)
/* UCS/WE = 12V, Up = 0V
A := E
OUT (Port A)
A := <HL>
OUT (Port B)
A := D
A := A v 90H
OUT (Port C)
/* UCS/WE = 12V, Up = 26V
A := 35H (10H)
/* ZS1, Startwert für tpw = 500 (150)µs
A := A - 1
A := 0
A := D
A := A v A0H
OUT (Port C)
/* UCS/WE = 12V, Up = 0V
DE := 03FFH
B := B - 1
B := 0
A := D
OUT (Port C)
/* UCS/WE = 0V, Up = 0V
RETURN
```

Bild 11: Unterprogramm PROGRAM

Literatur

- [1] Herrmann, W.: Technik der Mikrorechner. radio fernsehen elektronik 28 (1979) H. 2, S. 91 bis 94; H. 3, S. 163-166; H. 4, S. 227-230; H. 5, S. 301 und 302; H. 6, S. 367 und 368, 377 und 378
- [2] Schnupp, P.; Floyd, Ch.: SOFTWARE - Programmentwicklung und Projektorganisation. Berlin, New York: de Gruyter 1976
- [3] Kübart, R.; Niklisch, G.: PROM-Programmiergerät. radio fernsehen elektronik 29 (1980) H. 4, S. 225-227
- [4] Licht, U.: PROMs mit geringem Schaltungsaufwand programmiert. Elektronik 27 (1978) H. 8, S. 61-63

Tafel 4: Steuerung der Spannungen U_P und $U_{CS/WE}$ durch PC7 bis PC4 (Port C)

PC7 bis PC4	U_P	$U_{CS/WE}$
AH	0 V	12 V
9H	26 V	12 V
sonst	0 V	0 V

Bild 12: Unterprogramm FEHLER (Bild links)

Bild 13: Hauptprogramm PROMPRO 2 (Bild mitte)

Bild 14: Unterprogramm COMPARE (Bild rechts)

```
FEHLER
/* Soll-/Istwertvergleich mit Fehlerausgabe
A := 82H
OUT (Steuerlogik 8255)
/* Port B in Eingaberichtung
HL := XXXXH-1
DE := FFFFH
HL := HL + 1
DE := DE + 1
A := D
OUT (Port C)
/* UCE/WE = 0V, Up = 0V
A := E
OUT (Port A)
IN (Port B)
A := <HL>
0 Ausgabe (DE, A, <HL>)
DE := 03FFH
RETURN
```

12

```
PROMPRO 2
/* EPROM-Programmierung mit Vergleichstest
C := 10H
/* 16 Programmierversuche PV
C := C - 1
CALL, PROGRAM
/* N = 12, tpw = 150µs
CALL, COMPARE
/* Soll-/Istwertvergleich
Z = 1
C = 0
1 CALL, PROGRAM
2 CALL, FEHLER
END
```

13

```
COMPARE
/* Soll-/Istwertvergleich
A := 82H
OUT (Steuerlogik 8255)
/* Port B in Eingaberichtung
HL := XXXXH-1
DE := FFFFH
HL := HL + 1
DE := DE + 1
A := E
OUT (Port A)
A := D
OUT (Port C)
/* UCS/WE = 0V, Up = 0V
IN (Port B)
A := <HL>
DE := 03FFH
1 Z := 0
2 Z := 1
RETURN
```

14