

Bild 7.2 Power-off-Schaltung, Tongenerator

## 7.2. EPROM-Programmiergerät

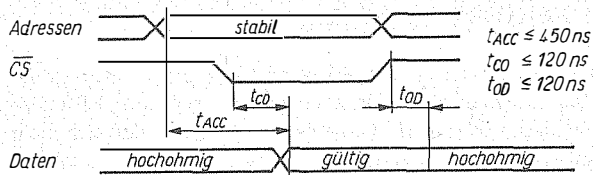
Ein wichtiger Zusatz zum Computer ist das EPROM-Programmiergerät. Da man zukünftig auch höher integrierte EPROM-Schaltkreise erhalten wird, wurde es so entwickelt, daß sich nicht nur Speicherschaltkreise vom Typ *U555* bzw. *2708* programmieren lassen, sondern das gesamte Typenspektrum: vom „Kleinsten“ *2704* bis zum *27256*. Die Anschlußbelegungen der verschiedenen EPROM-Typen sind im Anhang enthalten. Ein Vergleich der Pin-Belegungen zeigt, daß es nur wenige Unterschiede gibt. Bei EPROM-Schaltkreisen im 24poligen DIL-Gehäuse sind die Pins 18 ... 21 und 24. Über diese Pins werden  $\overline{CS}$ , Programmierspannung und Programmierimpulse, Betriebsspannungen und/oder weitere Adreßbit zugeführt. EPROM-Schaltkreise ab *2764* befinden sich im 28poligen Gehäuse. Bei diesen Schaltkreisen kommen noch 4 Anschlüsse hinzu, für die gleiches gilt. Die genannten Signale werden über Codierstecker zugeführt. Mit Hilfe dieser Codierstecker wird die Hardware an den jeweiligen EPROM-Typ angepaßt.

### 7.2.1. Programmiervorschriften

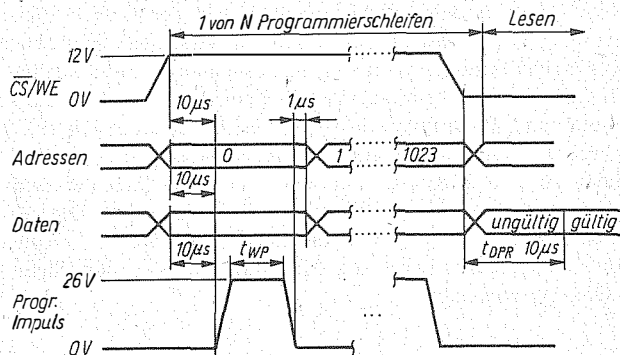
Im unprogrammierten Zustand des EPROM-Schaltkreises sind alle Bit 1 (Ausgänge H-Pegel). Programmiert wird somit durch Einschreiben von 0 Bit in die adressierten Zellen. Eine programmierte 0 kann nicht mehr unprogrammiert werden.

Bild 7.3 verdeutlicht das Programmieren des *U555* an Hand des Impulsdiagramms. Den  $\overline{CS}/\overline{WE}$ -Eingang legt man auf +12 V, anschließend die 1. Adresse und das zugehörige Datenwort (Ausgänge  $Q_0 \dots Q_7$ ) an den EPROM. Es folgt ein Programmierimpuls (0,1 ... 1 ms, 26 V). Danach wird die Adresse um 1 erhöht und der Vorgang wiederholt, bis alle Adressen abgearbeitet sind. Das muß *N*-mal für alle 1024 Adressen wiederholt werden. Die Anzahl der Programmierschleifen *N* hängt von der Breite des Programmierimpulses  $t_{pw}$  ab. Die integrale Programmierdauer je Speicherplatz beträgt 100 ms, so daß bei  $t_{wp} = 1$  ms 100 Programmierschleifen notwendig sind (*Intel*-Programmierschrift).

Moderne EPROM lassen sich einfacher programmieren. Der Schaltkreis *2716* wird mit 50-ms-TTL-Impulsen programmiert. Er



a)



b)

Bild 7.3 a – Impulsdiagramm U 555, b – Impulsdiagramm 2708

gestattet das Programmieren einzelner Speicherzellen und das Prüfllesen während des Programmiervorgangs. Das Programmieren kann folgendermaßen ablaufen: Anlegen der Programmierspannung  $U_{pp}$  (25 V) an Pin 21,  $\overline{OE} = H$ -Pegel, Anlegen Adresse und Datenbyte, Programmierimpuls (max. 55 ms) dem Eingang  $\overline{CE}$  zuführen. Ein Prüfllesen ist möglich, wenn der  $\overline{OE}$ -Eingang auf L gelegt wird, dabei muß  $U_{pp}$  nicht abgeschaltet werden. Bild 7.4 zeigt das Impulsdiagramm.

Der Schaltkreis 2732 wird ähnlich programmiert (Bild 7.5). Die Programmierspannung (25 V) wird dem Eingang  $\overline{OE}/V_{pp}$  zugeführt, der Programmierimpuls (50 ms; L-Pegel) dem Eingang  $\overline{CE}$ .

Die genauen Programmiervorschriften sind unbedingt den Datenblättern des Herstellers zu entnehmen. Beispielsweise wird für den U 555 eine integrale Programmierzeit von 50 ms je Speicherzelle

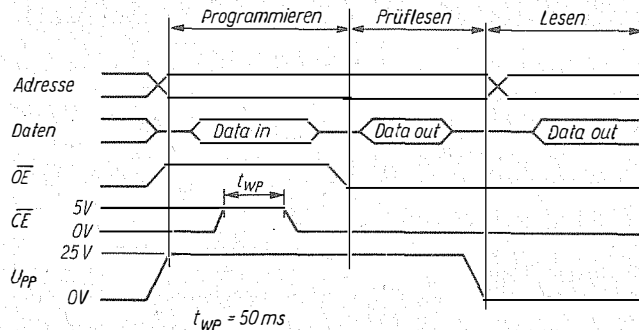


Bild 7.4 Impulsdiagramm 2716

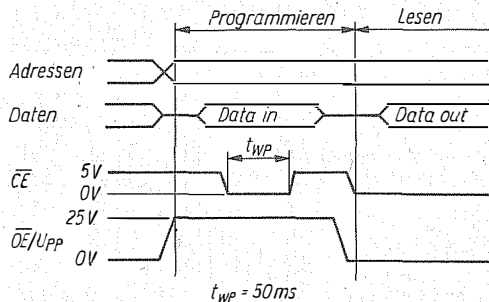


Bild 7.5 Impulsdiagramm 2732

[6], für den Intel 2708 aber 100 ms [7] angegeben. Ein weiterer Hinweis in diesem Zusammenhang: Die EPROM-Schaltkreise TMS 2716 und i 2716 sind nicht kompatibel.

### 7.2.2. Die Schaltung

Bild 7.6 zeigt die Schaltung des Programmiergeräts. Ein PIO-Baustein überträgt die Daten und Steuersignale von und zum Programmiergerät. Die Daten werden über Port A übertragen. Um einen PIO-Baustein einzusparen, erzeugt man die Adressen hardwaremäßig. Dafür sind die Zähler D1 ... D4 vorgesehen. Der Zähler

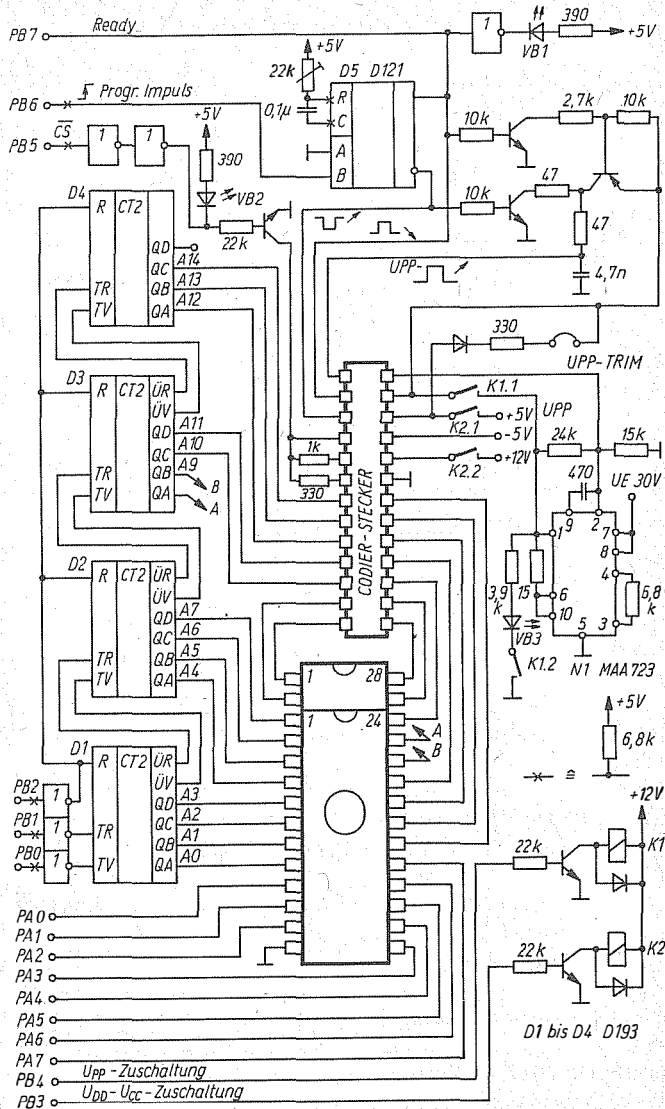


Bild 7.6 EPROM-Programmiergerät

D4 wird erst notwendig, wenn auch EPROM-Schaltkreise mit einer Kapazität > 4 kByte programmiert werden sollen. Die Zählerausgänge sind mit dem EPROM-Sockel (A0 bis A9) und dem Codierstecker (A10 bis A14) verbunden.

Den Programmierimpuls von 1 ms Länge erzeugt der Monoflop D5. Diese Variante bietet gegenüber einer softwaremäßigen Lösung den Vorteil, daß auch im Havariefall die Impulsbreite 1 ms nicht überschritten und so ein Zerstören des EPROM verhindert wird. Um bei der Programmierung der Typen 2716, 2732, 2764 usw. die notwendige Programmierzeit von 50 ms je Speicherplatz zu erreichen, wird der monostabile Multivibrator 50mal angestoßen. Die Ausgänge Q und  $\bar{Q}$  des Monoflop werden auf den Codierstecker geführt. Diese beiden Signale nutzt man bei den Schaltkreisen 2716, 2732 usw. als Programmierimpuls. Beim Programmieren der Typen U 555, 2708, TMS 2716 muß der 1-ms-Impuls auf etwa 26 V verstärkt werden. Dafür ist der Verstärker mit den Transistoren VT1 ... VT3 vorgesehen. Die 26-V-Impulse gelangen ebenfalls an den Codierstecker. Programmierimpulse zeigt die Leuchtdiode VB1 an.

Der Transistor VT4 arbeitet als Schaltstufe für das  $\overline{CS}/\overline{WE}$ -Signal beim U 555 bzw. 2708 oder für das  $\overline{OE}$ -Signal beim 2716 und 2732. Den Zustand dieses Signals zeigt die Leuchtdiode VB2 an. Mit dem Relais K1 wird die Programmierspannung zugeschaltet. Die Leuchtdiode VB3 hat Kontrollfunktion. Das Programmiergerät enthält ein Netzteil für die Programmierspannung  $U_{pp}$ . Die Spannung stabilisiert man mit dem Schaltkreis N1 (MAA 723). Der Knotenpunkt der Widerstände R1/R2 wird auf den Codierstecker geführt, um dort die genaue Programmierspannung mit einem Zusatzwiderstand festzulegen. Das Relais K2 schaltet die Betriebsspannung des EPROM-Schaltkreises softwaregesteuert zu.

### 7.2.3. Aufbau und Codierung

Die Programmierereinrichtung wurde als externes Gerät aufgebaut und läßt sich bei Bedarf über ein Kabel mit der PIO 2 verbinden. Das hat den Vorteil, daß die PIO 2 auch für andere Ein-/Ausgabearbeiten genutzt werden kann.

Als EPROM-Fassung sollte ein 28poliger Nullkraftsockel eingesetzt werden. Ein 26poliger direkter Steckverbinder nimmt die Co-

dierstecker auf. Sie bestehen aus 30 mm × 35 mm großen doppelt kaschierten Leiterplatten. Bild 7.7 zeigt die Anschlußbelegung der Codiereinrichtung. Bild 7.8 bis Bild 7.10 zeigen die Beschaltung der Pins 18 . . . 21 sowie die Verdrahtung der Codierstecker für die EPROM-Typen *U 555 (2708)*, *2716* und *2732* (vergleiche auch Bild 7.3 bis Bild 7.5).

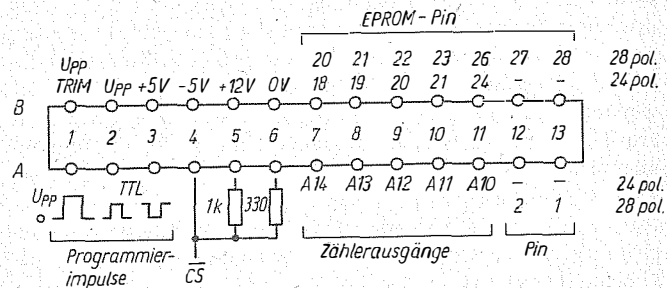


Bild 7.7 Anschlußbelegung der Codiereinrichtung

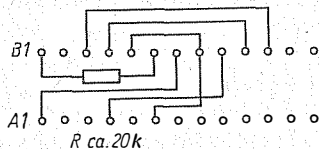


Bild 7.8 Codierstecker für *U 555 (2708)*

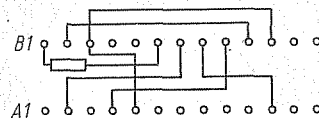


Bild 7.9 Codierstecker für *2716*

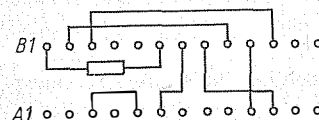


Bild 7.10 Codierstecker für *2732*

Die Verbindung des Programmiergeräts mit dem Computer übernimmt die PIO 2. Dabei sind die beiden Ports wie folgt belegt:

- Port A    Daten
- Port B    0 Vorwärtszählen (positiver Impuls)  
 1 Rückwärtszählen (positiver Impuls)  
 2 Rücksetzen des Zählers (negativer Impuls)  
 3 Zuschalten  $U_{cc}$  und  $U_{DD}$  (H)  
 4 Zuschalten  $U_{pp}$  (H)  
 5  $\overline{CS}/\overline{WE}$  bzw.  $\overline{OE}$  (H = Lesen)  
 6 Programmierimpuls (L/H-Flanke)  
 7 Fertigsignal (H für die Dauer des Impulses)

#### 7.2.4. Ein Beispielprogramm

An einem BASIC-geschriebenen Programm soll das Programmieren eines *U 555* demonstriert werden. Dabei wird abweichend von der *Intel*-Vorschrift programmiert [8], [9].

```

10 PRINT
20 PRINT „U 555/2708 PROGRAMMER“
30 PRINT
60 DA = 16: CA = 17: DB = 18: CB = 19: REM
   Ports festlegen
70 OUTCB, & CF: OUTCB, & 80: REM Bitmode, Bit
   7 Eingabe
80 OUTCA, & 7F: REM Byteeingabe
90 OUTDB, & 21: REM Spannungen aus, Zählerreset.
100 INPUT „LESEN (1) PROG (2) VERGLEICHEN (3)
   LÖSCHTEST (4)“: M
110 IF M > 4 THEN END
120 ON M GOSUB 250, 320, 550, 140
130 GOTO 90
140 REM Löschkontrolle
150 OUTCA, & 7F: REM Byteeingabe
160 OUTDB, & 29: OUTDB, & 2D: REM Zählerrest
170 F = 0: REM Flag
180 FOR I = 1 TO 1024: IF INP (DA) <> & FF THEN F =
   1: I = 1024
190 OUTDB, & 2c: OUTDB, & 2D: REM weiterzählen
200 NEXT I
  
```

```

210 PRINT „EPROM“; : IFF = 1 THEN PRINT „NICHT“;
220 PRINT „GELÖSCHT“ : RETURN
250 REM Dublizieren des EPROM-Inhalts ins RAM
260 INPUT „ADRESSE“; A
270 OUT CA, & 7F : REM Byteeingabe
280 OUT DB, & 29 : OUT DB, & 2D : REM Reset
290 FOR I = A TO A + 1023 : POKE I, INP (DA)
300 OUT DB, & 2c : OUT DB, & 2D : REM weiterzählen
310 NEXT I : RETURN
320 REM programmieren
330 INPUT „ADRESSE“; A : GOSUB 150 : IFF = 1 THEN
INPUT „WEITER (J/N)“; W$ : I : F W$ <> „J“ THEN
RETURN
340 PRINT : Z = 0 : REM Zykluszähler = 0
350 N = 2 : GOSUB 430 : REM 2 Programmiersversuche
360 REM Prüfllesen
370 OUT CA & 7F : OUT DB, & 2D : OUT DB, & 29 : OUT
DB, & 2D
380 F = 0 : FOR I = A TO A + 1023 : IF INP (DA) <> PEEK
(I) THEN F = 1 : I = A + 1024
390 OUT DB, & 2C : OUT DB, & 2D : NEXT I
400 IFF = 1 AND Z < 32 THEN 350
410 IFF = 1 THEN PRINT „NICHTPROGRAMMIERT“; :
GOTO 520
420 PRINT : N = Z/2 : GOSUB 430 : GOTO 520 : REM
Sicherheitszyklen
430 OUT DB, & 29 : OUT DB, & 19 : OUT DB, & 1D : REM
Zählerreset und Spannungen anlegen
440 OUT CA, & F : REM Byteausgabe
450 FOR I = 1 TO N : Z = Z + 1 : PRINT CHR$( &B);
USING „PROG ZYKLUS # # #“; Z
460 OUT DB, & 19 : OUT DB, & 1D : REM Zähler
rücksetzen
470 FOR I = A TO A + 1023 : OUT DA, PEEK (I)
475 REM Programmierimpuls erzeugen und weiterzählen
480 OUT DB, & 5D : OUT DB, & 1D : OUT DB, & 1C :
OUT DB, & 1D
490 NEXT I, J : RETURN
500 REM Auf Fertigsignal (Bit 7) wird nicht gewartet,
weil Schleifenlaufzeit > 1 ms

```

```

510 OUT CA & 7F : REM Byteeingabe
520 OUT DB, & 2D : REM Auf Lesen schalten
525 REM Prüfllesen, anschließend Signalton
530 GOSUB 560 : OUT 5, 0 : FOR I = 1 TO 500 : NEXT
I : OUT 4, 0
540 RETURN
550 INPUT „ADRESSE“; A
560 OUT CA, & 7F : REM Byteeingabe
570 OUT DB, & 29 : OUT DB, & 2D : REM Zähler
rücksetzen
580 E = 0 : FOR I = A TO A + 1023
590 IF INP (DA) <> PEEK (I) THEN E = E + 1
600 OUT DB, & 2C : OUT DB, & 2D : REM weiterzählen
610 NEXT I : PRINT E; „FEHLER“ : RETURN

```

### 7.3. Analogschnittstellen

Will man mit dem Computer auch Analogsignale oder Meßwerte erfassen und verarbeiten, so sind geeignete Analog/Digital-Wandler und Digital/Analog-Wandler erforderlich. In diesem Abschnitt werden einige Analogschnittstellen beschrieben.

#### 7.3.1. Digital/Analog-Wandler

International gibt es eine große Anzahl hybrider und monolithischer D/A-Wandlerbausteine. Ein besonders günstiger Baustein ist der sowjetische CMOS-Wandler *K 572 ИА1*. Es handelt sich bei diesem Schaltkreis um einen TTL-kompatiblen multiplizierenden 10-Bit-Wandler mit einer Einschwingzeit von 500 ns. Die Betriebsspannung kann 5 . . . 15 V ( $P_{\text{vmax}} = 20 \text{ mW}$ ), die Referenzspannung maximal  $\pm 10 \text{ V}$  betragen. Bild 7.11 zeigt das Funktionsprinzip des *K 572 ИА1* und Bild 7.12 die Standardbeschaltung. Die Ausgangsspannung beträgt

$$U_A = U_{\text{ref}} \sum_{i=1}^n \frac{S_i}{2^{n-i+1}}$$

Dabei gilt:  $S_i = 1$  bei H und  $S_i = 0$  bei L am Digitaleingang  $E_i$  und  $n = 10$ .

Bild 7.13 zeigt die Beschaltung des D/A-Wandlers für bipolare