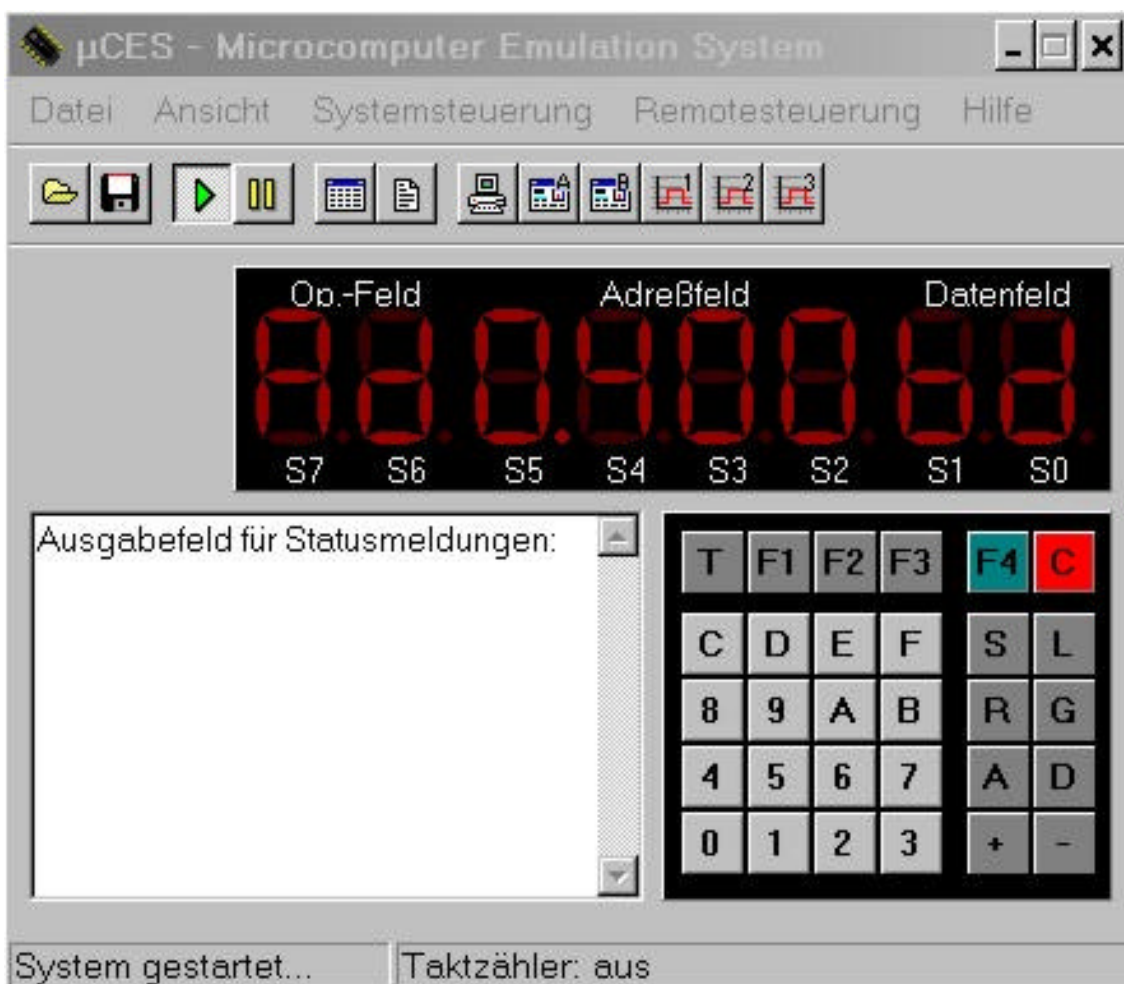


# Skript zum Mikrorechner-Praktikum

## Kapitel 5:

### Aufbau und Funktion der Schnittstellenbausteine - Teil II

Autor: Helmut Bähring



## **Inhalt des Kapitels 5:**

5. Aufbau und Funktion der Schnittstellenbausteine - Teil II	1
5.1 Der Zeitgeber-/Zähler-Baustein MC6840	1
5.1.1 Aufbau des Bausteins	1
5.1.2 Die Funktionen	5
5.1.3 Der Registersatz	8
5.2 Praktische Übungen zum Zeitgeber-/Zählerbaustein	11
5.2.1 Programmierbarer Rechteckgenerator mit dem Zähler #3	11
5.2.2 Zykluszeit-/Impulsdauermessung	15
Lösungsvorschläge zu den Praktischen Übungen	19
Anhang: Der Registersatz des MC6840	29

## 5 Aufbau und Funktion der Schnittstellenbausteine - Teil II

### 5.1 Der Zeitgeber-/Zähler-Baustein MC6840

#### 5.1.1 Aufbau des Bausteins

Im Praktikumsrechner ist als Zeitgeber-/Zähler-Baustein<sup>1</sup> der Typ MC6840 eingesetzt, der von der Herstellerfirma Motorola mit dem Namen

*Programmable Timer Module (PTM)*

bezeichnet wird. Er stellt drei unabhängig arbeitende 16-bit-Zähler zur Verfügung. Jeder Zähler besitzt ein Steuerregister (CR), einen externen Takteingang ( $\overline{C}$ ), einen Steuereingang ( $\overline{G}$ ) und einen Impulsausgang (O). Beim Erreichen des Zählerwertes 0 (*time out*) wird ein Bit in einem zentralen Statusregister gesetzt und (wahlweise) der Prozessor durch einen Interrupt davon unterrichtet. Über den Zählerausgang können Rechteckschwingungen mit fester oder variabler (modulierter) Impulsbreite oder einzelne Impulse bestimmter Länge ausgegeben werden. Die Einsatzgebiete des Bausteins sind:

- Erzeugung von Signalen zur Zeitmessung (*Wave Synthesis Mode*, z.B. Echtzeit-Uhr),
- Frequenzmessung (*Wave Measurement Mode*),
- Intervallängenmessung (*Wave Measurement Mode*),
- Ereigniszählung.

Der Baustein arbeitet typabhängig mit einer maximalen internen Frequenz von 1 bis 2 MHz (Takt E des MC6809). Die zulässige minimale Frequenz des Systemtaktes ist 100 kHz. Im Bild 5.1-1 ist der Aufbau eines einzelnen Zählers dargestellt.

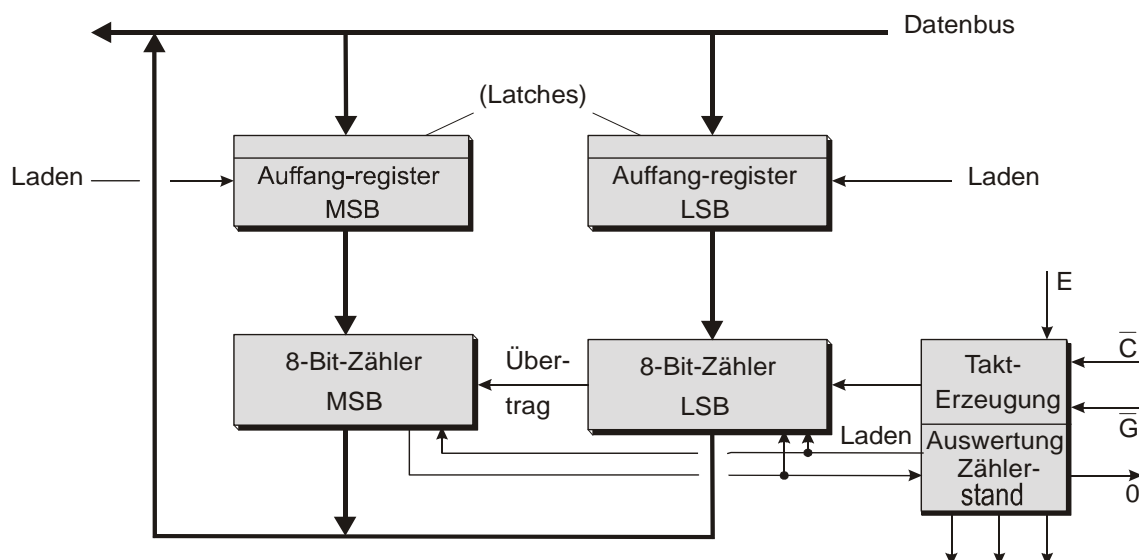


Bild 5.1-1: Aufbau eines der 16-bit-Zähler im MC6840

<sup>1</sup> Zeitgeber-/Zählerbausteine werden ausführlich in Abschnitt II.3.4 beschrieben.

Jeder Zähler besteht aus zwei 8-bit-Zählern (MSB - *Most Significant Byte*, LSB - *Least Significant Byte*), die cascadiert als 16-bit-Zähler oder als zwei getrennte 8-bit-Zähler arbeiten können. Jedem Zähler ist ein Auffangregister (*Latch*) vorgeschaltet, das über den Datenbus geladen wird. Durch ein Signal wird der Zustand der Auffangregister parallel in die 8-bit-Zähler übernommen. Dieser Vorgang wird mit **Initialisierung** des Zählers bezeichnet. Durch den Steuereingang  $\overline{G}$  (*Gate*) kann der Zählvorgang gestartet oder abgebrochen werden. Mit dem Starten beginnt der **Zählzyklus**, wobei wahlweise der interne Takt E oder ein externer Takt  $\overline{C}$  benutzt werden kann. Im **16-bit-Zählmodus** wird nach Erreichen des Wertes 0 (*time out*) der Zähler mit dem nächsten Takt erneut aus seinen Auffangregistern geladen. Im **2x8-bit-Zählmodus** wird jedesmal, wenn der LSB-Zähler den Wert 0 erreicht, dieser aus seinem Auffangregister geladen und der höherwertige um 1 dekrementiert. Erst wenn beide Zähler den Wert 0 erreichen, werden sie gleichzeitig aus ihren Auffangregistern initialisiert. Der Begriff **Zählzyklus** bezeichnet immer die Zeit zwischen zwei Initialisierungen des Zählers. Durch eine Schaltung, die den Zählerstand auswertet, wird ein digitales Signal einer wählbaren Form auf den Ausgang O gegeben. Auf diese Signale wird zum Schluß dieses Abschnittes genauer eingegangen.

Das Blockschaltbild Bild 5.1-2 zeigt den inneren Aufbau des Bausteins MC6840.

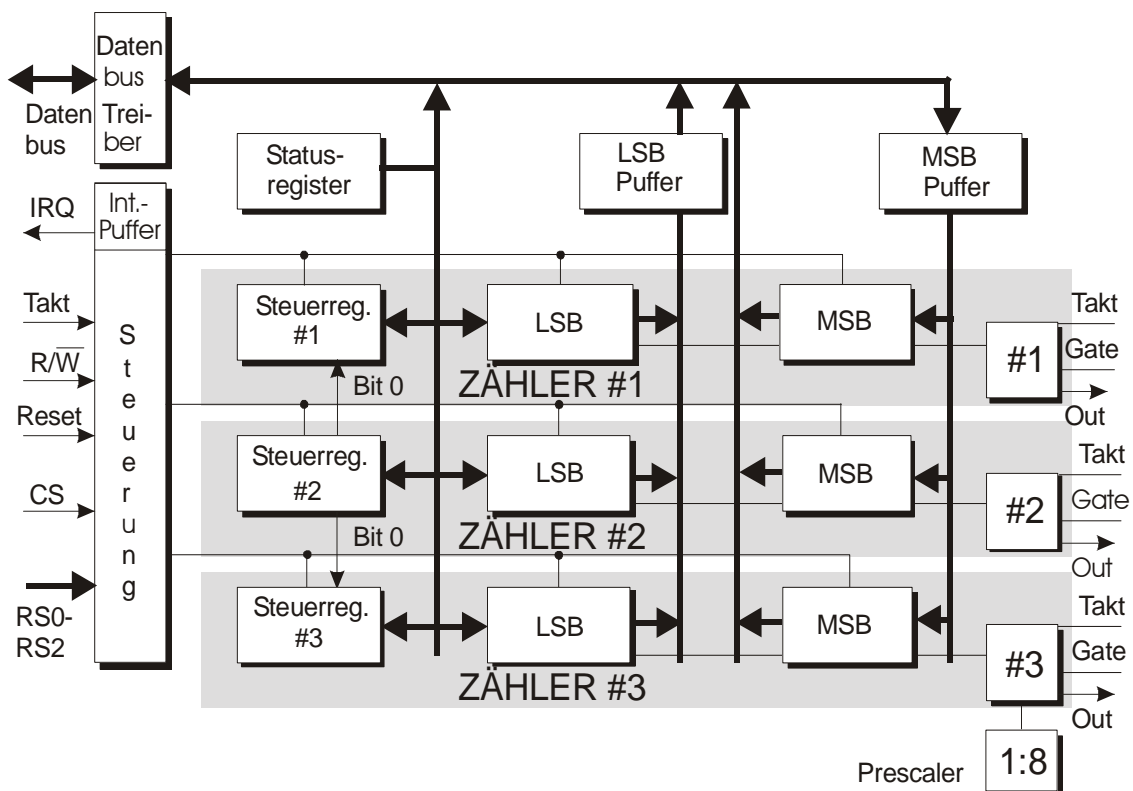


Bild 5.1-2: Blockschaltbild des Bausteins MC6840

Die Schnittstelle zum Prozessor weist keinerlei Besonderheiten gegenüber den im Kapitel 4 beschriebenen Bausteinen auf. Es stehen zwei Eingänge zur Bausteinauswahl ( $\overline{CS0}$ ,  $CS1$ ) und drei Eingänge zur Auswahl der internen Register ( $RS0$ ,  $RS1$ ,  $RS2$ ) zur Verfügung. Dem Gesamtplan des Rechners im Anhang des Kapitels 3 können

Sie entnehmen, daß die Selektion des Bausteins durch das Signal  $\overline{CS12}$  und die Adreßleitung A3 im Adreßbereich \$F018 - \$F01F geschieht (vgl. Selbsttestaufgabe S3.1-2 in Kapitel 3). Die Schnittstelle zur Peripherie besteht für jeden Zähler #i aus dem schon erwähnten Takteingang  $\overline{C_i}$ , dem Steuereingang  $\overline{G_i}$  und dem Signalausgang  $O_i$ . Durch eine Zusatzschaltung (*Prescaler*) kann die Taktfrequenz des Zählers #3 auf 1/8 herabgesetzt werden.

Im Blockschaltbild Bild 5.1-2 erkennt man, daß nur drei Selektionssignale RS2-RS0 für die Auswahl von mehr als acht Registern zur Verfügung stehen. Dies wird dadurch ermöglicht, daß teilweise mehrere Register unter der gleichen Adresse angesprochen werden:

- Da das Statusregister nur gelesen, die Steuerregister nur beschrieben werden, konnte man das Statusregister und das Steuerregister von Zähler #2 mit der gleichen Adresse versehen. Das  $R/\overline{W}$ -Signal wird zur Selektion genau eines der beiden Register mit herangezogen.
- Die Steuerregister der Zähler #1 und #3 besitzen die gleiche Adresse. Die Selektion unter ihnen wird durch das Bit 0 des Steuerregisters des Zählers #2 vorgenommen, das somit als zusätzliches Adreßbit benutzt wird (vgl. Bit CRX2 in der PIA MC6821, Kapitel 4, Abschnitt 4.2).

Bild 5.1-3 zeigt die Anschlußbelegung des Bausteins MC6840.

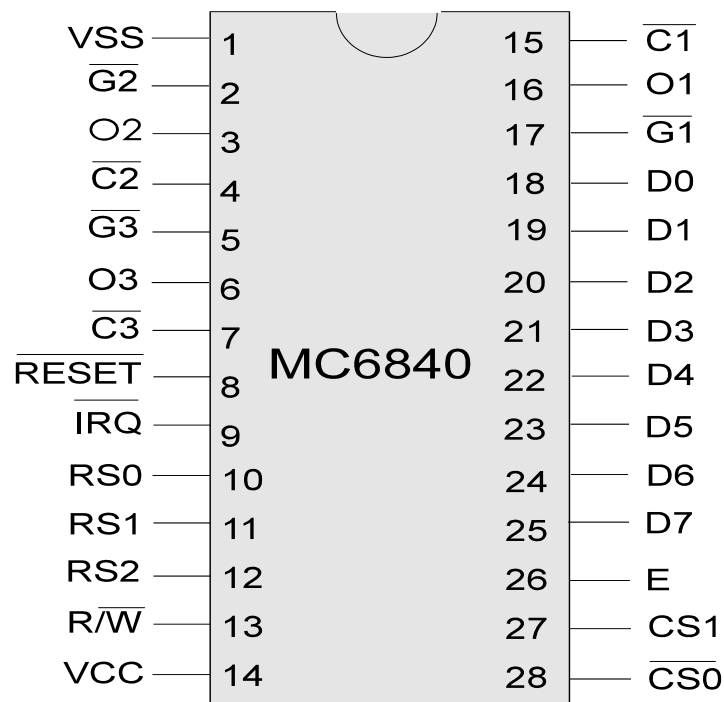


Bild 5.1-3: Anschlußbelegung des Bausteins MC6840

Tabelle 5.1-1 gibt die Adreßzuteilung der Register des MC6840 im Praktikumsrechner wieder. Darin bezeichnen CR#1, CR#2, CR#3 die Steuerregister der Zähler #1, #2, #3 und "Latches #i" die Auffangregister des Zählers #i.

Tabelle 5.1-1: Adreßzuteilung der Register des MC6840

Adresse	RS2	RS1	RS0	Schreiben R/W=0)	(Lesen(R/W=1)
\$F018	0	0	0	CR#3 (Bit0=0 im CR#2)	-
\$F018	0	0	0	CR#1 (Bit0=1 im CR#2)	-
\$F019	0	0	1	CR#2	Statusregister
\$F01A	0	1	0	MSB-Pufferregister	MSB-Zähler #1
\$F01B	0	1	1	Latches #1	LSB-Puffer
\$F01C	1	0	0	MSB-Pufferregister	MSB-Zähler #2
\$F01D	1	0	1	Latches #2	LSB-Puffer
\$F01E	1	1	0	MSB-Pufferregister	MSB-Zähler #3
\$F01F	1	1	1	Latches #3	LSB-Puffer

Weiter oben wurde bereits die Funktion der Zähler beschrieben. Da beim Erreichen des Null-Wertes die Auffangregister gleichzeitig in die 8-bit-Zähler geladen werden, ist es wichtig, vom Prozessor gewünschte Änderungen der Auffangregister gleichzeitig in allen 16 Bits vorzunehmen. Die LSB-Zähler besitzen dazu ebenso wie die MSB-Zähler jeweils einen getrennten Eingabebus und Ausgabebus. Der Ausgabebus der LSB-Zähler und der Eingabebus der MSB-Zähler werden über spezielle Puffer(register) geführt. In Tabelle 5.1-1 erkennt man, daß diese Register jeweils unter drei verschiedenen Adressen angesprochen werden. Beispielhaft wird nun die Initialisierung des Zählers #2 erklärt:

Aus Kapitel 1, Abschnitt 1.3, wissen Sie, daß der Prozessor MC6809 den speziellen Befehl STD besitzt, mit dem der 16-bit-Akkumulator D in zwei aufeinander folgenden Speicherzellen abgelegt werden kann. Dem Befehl zum Abspeichern von D in den Auffangregistern von Zähler #2

STD \$F01C                      entspricht die Befehlsfolge

1. STA \$F01C
2. STB \$F01D

Bei der Ausführung dieser Folge im MC6840 wird im

1. Schritt: der Akkumulator A in das MSB-Pufferregister übertragen,
2. Schritt: der Akkumulator B in das Auffangregister des LSB-Zählers #2 und gleichzeitig das MSB-Pufferregister in das Auffangregister des MSB-Zählers #2 übertragen.

Ausführung dieser Befehle in umgekehrter Reihenfolge führt zu einem falschen Initialisierungswert, da nicht der aktuelle Wert in A, sondern ein zufällig vorher im MSB-Puffer gespeicherter Wert ins Latch übertragen wird. Auch Befehle, die Speicherzellen (oder Peripherieregister) modifizieren, z.B. INC, führen zu falschen Ergebnissen.

Beim Lesen des Zählerstandes durch den Prozessor ist es wichtig, daß er beide 8-bit-Zähler zum gleichen Zeitpunkt liest, da sonst durch zwischenzeitlich eintreffende Taktsignale der LSB-Zähler bereits seinen Zustand geändert haben könnte.

Das Lesen wird nun beispielhaft für den Zähler #3 vorgeführt.

Dem Befehl LDD \$F01E entspricht die Befehlsfolge

1. LDA \$F01E
2. LDB \$F01F

Bei der Ausführung dieser Folge im MC6840 wird im

1. Schritt: der Wert des MSB-Zählers #3 in den Akkumulator A und gleichzeitig der Wert des LSB-Zählers #3 in das LSB-Pufferregister geladen.
2. Schritt: der Wert des LSB-Pufferregisters in den Akkumulator B übertragen.

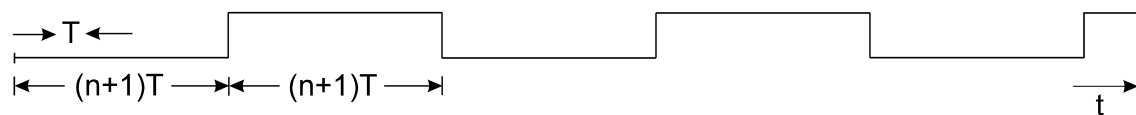
### 5.1.2 Die Funktionen

Weiter oben wurden bereits die Betriebsarten beschrieben, in denen die Zähler arbeiten können:

- Im **16-bit-Modus** besteht ein Zählzyklus aus  $(n+1)$  Taktzyklen, wobei  $n$ ,  $0 \leq n < 2^{16}$ , der Wert der Auffangregister des Zählers ist.
- Im **2x8-bit-Modus** besteht ein Zählzyklus aus  $(k+1)(m+1)$  Taktzyklen, wobei  $k$  und  $m$  die Werte des LSB- bzw. MSB-Auffangregisters des Zählers sind ( $0 \leq k, m < 2^8$ ). Dabei sind zwei Spezialfälle zu berücksichtigen:
  - $k=0$ : Der Ausgang O wechselt nach jedem Zählzyklus (der Länge  $m+1$  Taktzyklen) seinen Zustand. Also wird ein symmetrisches Rechtecksignal (*square wave*) erzeugt.
  - $k=m=0$ : Am Ausgang O erscheint ein Square Wave-Signal mit der halben Taktfrequenz des Zählers.
- Der Zähler arbeitet als sog. **Timer**, wenn er mit dem internen oder einem periodischen externen Takt betrieben wird.
- Der Zähler arbeitet als **Ereigniszähler**, wenn er mit einem externen (meist aperiodischen) Signal am Eingang  $\overline{C}$  getaktet wird.
- Der Zähler arbeitet im **zyklischen Zählmodus**, wenn sich am Ausgang O das Signal mit jedem Zählzyklus periodisch wiederholt.
- Der Zähler arbeitet im **Single-Shot-Modus (Monoflop-Betrieb)**, wenn das Signal am Ausgang O einmalig nur während des 1. Zählzyklus nach der Initialisierung des Zählers erscheint. Dabei läuft der Zähler intern kontinuierlich weiter und erzeugt (falls so programmiert) weiterhin Interrupts beim Nulldurchgang.

Im Bild 5.1-4 sind die Ausgangssignale für die genannten Zählmodi dargestellt. Die Zahlen  $n, m, k$  haben dabei die oben angegebene Bedeutung. Zu beachten ist noch, daß der Takteingang  $\overline{C}$  mit der negativen Flanke von E ausgewertet wird und daher die externe Frequenz an  $\overline{C}$  kleiner als die halbe Frequenz von E sein soll.

## Zyklischer Zählbetrieb: 16-Bit-Modus (Squarewave)



## 2 x 8-Bit-Modus:



## Monoflop-Zählbetrieb: 16-Bit-Modus



## 2 x 8-Bit-Modus:



Bild 5.1-4: Die Ausgangssignale in den verschiedenen Betriebsarten

Der externe Eingang  $\overline{G}$  (Gate) jedes Zählers wird bei jeder negativen Flanke des Taktsignals E abgefragt und kann zwei verschiedene Funktionen wahrnehmen:

- Als **Steuereingang** veranlaßt jede negative Flanke an  $\overline{G}$  die Initialisierung, d.h. das Laden des Zählers aus seinen Auffangregistern. Im zyklischen Zählmodus wird der Zähler nur solange dekrementiert, wie am Eingang  $\overline{G}$  ein L-Signal anliegt. Im Single-Shot-Modus bleibt eine Änderung des Gate-Signals zum H-Pegel ohne Auswirkung.
- Als **Vergleichseingang** erlaubt er, ein an diesem Eingang angelegtes digitales Signal zeitlich mit dem Zählzyklus zu vergleichen. Dabei wird der Zähler ebenfalls durch eine negative Flanke am Eingang  $\overline{G}$  initialisiert und gestartet.

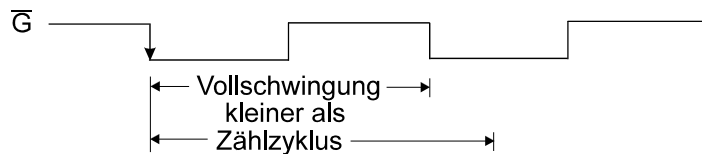
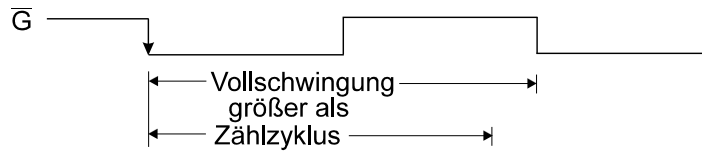
Ist der Eingang  $\overline{G}$  als Vergleichseingang (*Wave Measurement Mode*) programmiert, so können die folgenden zwei Vergleiche durchgeführt werden.

- Im **Frequenz-Vergleichsmodus** wird die Dauer des Zählzyklus mit der Dauer einer Vollschwingung des Signals am Gate-Eingang  $\overline{G}$  verglichen.
- Im **Pulsbreiten-Vergleichsmodus** wird die Dauer des Zählzyklus mit der Breite des negativen Impulses des Signals am Eingang  $\overline{G}$  verglichen.



Das Bild 5.1-5 zeigt die vier möglichen Ergebnisse des Vergleiches.

#### Frequenzvergleich



#### Pulsbreitenvergleich

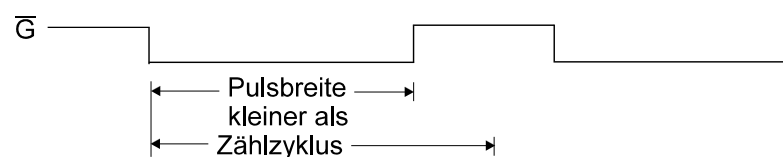
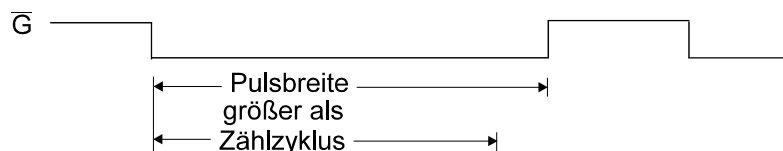


Bild 5.1-5: Zum Vergleich des Zählzyklus mit einem externen Signal

Wahlweise führt das Ergebnis

- Zählzyklus > Vollschwingung oder
- Zählzyklus < Vollschwingung

im Frequenzvergleich,

- Zählzyklus > Impulsbreite oder
- Zählzyklus < Impulsbreite

im Pulsbreitenvergleich

zum Setzen des Interrupt-Flags im Statusregister und damit gegebenenfalls zu einer Unterbrechungsanforderung an den Prozessor.

Das letzte Unterscheidungsmerkmal der Betriebsarten besteht in den Bedingungen, die zu einer Initialisierung der Zähler, d.h. dem Laden aus ihren Auffangregistern führt.

- Im Frequenz- bzw. Pulsbreiten-Vergleichsmodus geschieht dies jeweils durch eine negative Flanke am Eingang  $\overline{G}$ .

- Im zyklischen oder Single-Shot-Betrieb geschieht dies ebenfalls durch eine negative Flanke am Eingang  $\overline{G}$ . Darüberhinaus wird es aber auch durch ein L-Signal am externen Rücksetzeingang (*Reset*) oder aber programmgesteuert (internes Rücksetzen, vgl. Bit 0 im Steuerregister CR#1, s.u.) veranlaßt. Zusätzlich kann gewählt werden, ob auch durch das Einschreiben eines neuen Wertes in die Auffangregister der Zähler eine Initialisierung durchgeführt wird oder nicht.

### 5.1.3 Der Registersatz

#### Das Statusregister

Im Bild 5.1-6 ist das zentrale Statusregister (SR) dargestellt, das für alle drei Zähler zuständig ist.

Bit	7	6	5	4	3	2	1	0
	INT	nicht benutzt				INT#3	INT#2	INT#1

Bild 5.1-6: Das Statusregister

(Adresse: \$F019)

Die drei unteren Bits sind als Interrupt-Flags den einzelnen Zählern zugeordnet: INT#i ist Interrupt-Flag von Zähler #i. Dieses Bit wird gesetzt, wenn der Zähler #i abgelaufen ist - unabhängig davon, ob eine Interruptanforderung für den Zähler #i im Steuerregister CR#i (s.u.) aktiviert ist oder nicht. Das Bit 7 (INT) wird als logische Verknüpfung aus den unteren drei Bits gewonnen. Es nimmt den logischen Wert 1 an, sobald eines dieser Bits und im Steuerwort des zugehörigen Zählers das *Interrupt Enable Bit 6* (s.u.) gesetzt ist. Anhand des Bits 7 kann der Prozessor zunächst entscheiden, ob vom Baustein MC6840 überhaupt eine Programmunterbrechung angefordert wird. Anhand der Bits 0-2 (INT#i) kann er dann genauer feststellen, welcher Zähler diesen Interrupt wünscht. Das Rücksetzen des Interrupt-Flags eines Zählers geschieht durch eine der folgenden Aktionen:

- ein L-Signal am Reset-Eingang des Bausteins,
- programmgesteuert durch das Bit 0 im Steuerregister CR#1 (s.u., "internes Rücksetzen")
- durch das Lesen des Statusregisters (während das Interrupt-Flag gesetzt ist) und nachfolgendem Lesen des Wertes des Zählers,
- durch das Einschreiben eines neuen Wertes in die Auffangregister des Zählers.

Die Interruptvektoren zu den Timern #1 - #3 des MC6840 finden Sie in der Tabelle im Anhang A zu Kapitel 3.

#### Das Steuerregister

In der folgenden Beschreibung der Funktion der Bits des Steuerregisters werden die Initialisierungen, die ohne Wahlmöglichkeiten stets durchgeführt werden, nicht mehr explizit aufgeführt. Im Bild 5.1-7 ist der Aufbau des Steuerregisters dargestellt.

Bit	7	6	5	4	3	2	1	0
	OE	IE	Modus, Interr.-Steuerung		ZM	T		

Bild 5.1-7: Die Steuerregister CR#i des MC6840

■ **Bit 7: (Output-Enable-Bit)**

Bit7=0: Ausgang O ist gesperrt, d.h. er liegt statisch auf L-Potential,  
 Bit7=1: Ausgang O aktiviert.

■ **Bit 6: (Interrupt-Enable-Bit)**

Bit6=0: Interrupt vom Zähler gesperrt (*disabled*),  
 Bit6=1: Interrupt vom Zähler zugelassen (*enabled*).

■ **Bits 5-3: (Ausgabe-Modus und Interrupt-Steuerung)**

**5 4 3**

- 0 0 0 Zyklischer Zählbetrieb, Initialisierung durch Schreiben der Latches.
- 0 0 1 Frequenzvergleich: Interrupt, falls Zählzyklus > Vollschwingung an  $\overline{G}$ ,  
Initialisierung durch Schreiben der Latches.
- 0 1 0 zyklischer Zählbetrieb, keine Initialisierung durch Schreiben der Latches.
- 0 1 1 Pulsbreitenvergleich: Interrupt, falls Zählzyklus > Pulsbreite an  $\overline{G}$ ,  
keine Initialisierung durch Schreiben der Latches.
- 1 0 0 Single-Shot-Betrieb (Monoflop-Betrieb), Initialisierung durch Schreiben  
der Latches.
- 1 0 1 Frequenzvergleich: Interrupt, falls Zählzyklus < Vollschwingung an  $\overline{G}$ ,  
Initialisierung durch Schreiben der Latches.
- 1 1 0 Single-Shot-Betrieb, keine Initialisierung durch Schreiben der Latches.
- 1 1 1 Pulsbreitenvergleich: Interrupt, falls Zählzyklus < Pulsbreite an  $\overline{G}$ ,  
keine Initialisierung durch Schreiben der Latches.

(Hinweis: Bit 4 steuert die Initialisierung durch Schreiben der Latches.)

■ **Bit 2: (Zählmodus-Auswahlbit)**

Bit2=0: 16-bit-Zählmodus,  
 Bit2=1: 2x8-bit-Zählmodus.

■ **Bit 1: (Taktauswahl-Bit)**

Bit1=0: Zähler wird extern über den Eingang C getaktet,  
 Bit1=1: Zähler wird intern vom Systemtakt E getaktet.

### ■ Bit 0:

Das Bit 0 hat in allen drei Steuerregistern CR#1, CR#2, CR#3 völlig unterschiedliche Funktionen:

- Steuerregister CR#1: **internes Rücksetzbit**

Bit0=0: alle Zähler sind aktiviert,

Bit0=1: alle Zähler werden aus ihren Auffangregistern initialisiert (internes Rücksetzen).

- Steuerregister CR#2: **Steuerregister-Adreßbit**

Bit0=0: CR#3 zum Schreiben vorselektiert,

Bit0=1: CR#1 zum Schreiben vorselektiert.

Steuerregister CR#3: **Taktsteuerung für Zähler #3**

Bit0=0: Takt wird unverändert zum Zählen benutzt,

Bit0=1: Takt wird auf 1/8 seiner Frequenz heruntergeteilt.

Zum Schluß dieses Abschnittes folgen noch einige Bemerkungen zum Rücksetzen des Bausteins.

### ■ Internes Rücksetzen:

programmierbar durch Rücksetzen des Bit 0 im Steuerregister CR#1 des Zählers #1 (s.o.). Alle Zähler werden aus ihren Auffangregistern geladen, der Wert der Steuerregister bleibt unverändert.

### ■ Externes Rücksetzen:

durch ein L-Signal am Eingang RESET des MC6840.

- Alle Zähler und ihre Auffangregister werden mit dem Wert \$FFFF geladen,
- alle Steuerregister und das Statusregister werden auf den Wert \$00 gesetzt, außer Bit 0 in CR#1, d.h. alle Zähler sind deaktiviert!
- alle Zählerausgänge (O1,O2,O3) und externen Takteingänge (C1,C2,C3) werden gesperrt.

## 5.2 Praktische Übungen zum Zeitgeber-/Zähler-Baustein

### 5.2.1 Programmierbarer Rechteckgenerator mit dem Zähler #3

In den folgenden Praktischen Übungen sollen Sie mit dem Baustein MC6840 Ausgangssignale erzeugen oder Eingangssignale ausmessen. Natürlich wäre es sehr viel besser, wenn Sie dazu auf einen Oszilloskop zurückgreifen könnten, auf dem Sie die Signale darstellen und vergleichen könnten. Leider müssen Sie sich mit der beschränkten Funktionalität der Schnittstellen-Erweiterungskarte, wie wir sie in Kapitel 4 beschrieben haben, zufrieden geben. Die Aufgaben, die ein Eingangssignal erfordern, können dabei nur auf dem Praktikumsrechner, nicht mit dem 6809-Simulator gelöst werden.

#### Praktische Übung P5.2-1:

Schließen Sie zunächst die Schnittstellen-Erweiterungskarte an den Praktikumsrechner an. Führen Sie die folgende Übung mit dem Praktikumsrechner und dem 6809-Simulator durch.

I. Programmieren Sie den Zähler #3 mit den folgenden Parametern:

1. Ausgang O des Zählers gesperrt,
2. Interrupt zum Prozessor aktiviert,
3. zyklischer Zählbetrieb mit Initialisierung der Zähler durchs Beschreiben der Auffangregister,
4. 16-bit-Zählmodus,
5. interner Zähltakt E,
6. auf 1/8 herabgesetzter Zähltakt.

II. Schreiben Sie nun den Wert \$FFFF in das Auffangregister des Zählers #3.

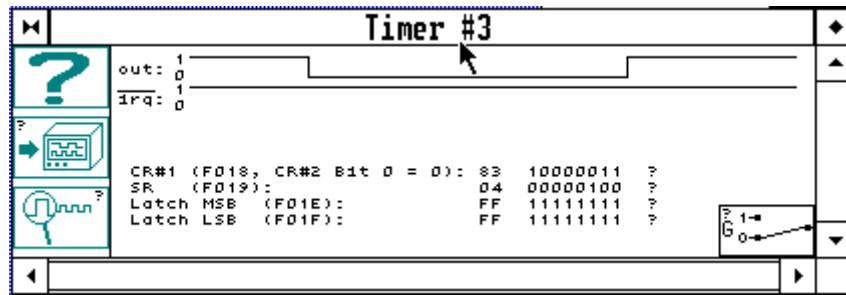
III. Belegen Sie nun den Interruptvektor zum Zähler #3 mit dem Wert \$E457, der Startadresse der Break-Routine im Monitor.

IV. Starten Sie den Zähler durch Löschen des Bit 0 im Steuerregister des Zählers #1. Beobachten Sie, was in der Anzeige des Rechners passiert und versuchen Sie eine Erklärung des Vorgangs. Nach welcher Zeit passiert dort etwas ?

V. Schreiben Sie ein Programm, das zunächst den Zähler wie unter I. gefordert programmiert, wobei aber zusätzlich der Ausgang O3 aktiviert werden soll. Ergänzen Sie das Programm um eine Interruptroutine, die einen 2stelligen Hexadezimalzähler in den Anzeigestellen S1,S0 mit jedem Interrupt des Zählers #3 hochzählt. Es soll die minimale Zählfrequenz erzeugt werden. Beobachten Sie den Ausgang O3.

### Zur Arbeit mit dem Emulator:

Öffnen Sie das Emulator-Fenster zum gewünschten Zeitgeber (Menüpunkt: Bausteine, Eintrag: Timer #i). Im unteren Teil des Fensters sehen Sie die aktuelle Belegung der Timer-Register. Der obere Teil wird von zwei Zeitsignalen eingenommen, die das Ausgangssignal  $O_i$  des Timers sowie den Interruptausgang darstellen. Durch Anklicken des Schalter-Symbols in der unteren, rechten Fensterecke können Sie das Gate des Timers betätigen und damit seine Funktion ein- oder ausschalten.



Nachdem ein Timer gestartet wurde, können Sie durch Anklicken der Ikone



die Ausgabe des Ausgangssignals  $O_i$  unterbrechen und den aufgezeichneten Signallauf (wie auf einem Oszilloskop) betrachten. Die Zeiteinheit auf der X-Achse ist dabei durch die abgelaufenen Taktzyklen seit der Aktivierung des Zählers gegeben. Der waagerechte Verschiebepalken erlaubt es, alle Zeitbereiche zu betrachten, mit Hilfe des senkrechten Verschiebepalkens kann die Auflösung der Zeitdarstellung verändert werden.

Das Anklicken der Ikone



startet wieder die Ausgabe der Ausgangssignale.

### Praktische Übung P5.2-2: Rechteckgenerator mit interner Taktung

Schließen Sie zunächst die Schnittstellen-Erweiterungskarte an den Praktikumsrechner an. Führen Sie die folgende Übung mit dem Praktikumsrechner und dem 6809-Simulator durch.

- I. Mit welchem Hexadezimalwert muß das Steuerregister CR#3 geladen werden, um folgende Betriebsart zu erhalten ?
  1. Ausgang 03 aktiviert,
  2. Interrupt zur CPU gesperrt,
  3. zyklischer Zählbetrieb,
  4. 16-bit-Zählmodus,
  5. Takt wird auf 1/8 heruntergeteilt,
  6. Taktauswahl intern, vom Systemtakt E (Frequenz 1MHz) abgeleitet.

- II. Welche Frequenz und Periodendauer wird das Rechtecksignal am Ausgang O3 haben, wenn in das Auffangregister von Zähler #3 nacheinander folgende Werte geladen werden? (Bitte geben Sie die Rechnungen an und zeichnen Sie die „Oszilloskop“-Bilder.)
- a) \$1000   b) \$8000   c) \$F000
- III. „Programmieren“<sup>1</sup> Sie den Baustein MC6840 des Praktikumsrechners für die unter I. genannte Funktionsweise und betrachten Sie das Signal am Ausgang O3 für die unter a)-c) angegebenen Initialwerte des Auffangregisters von Zähler #3. Vergessen Sie nicht den Zähler zu starten ! (Hilfe: Praktische Übung P5.2-1)
- IV. Welche Wirkung haben eine Betätigung des Gate-Eingangs  $\overline{G3}$ , das Einschreiben eines neuen Wertes in die Auffangregister sowie die Änderung des Steuerwortes in CR#3 während des Betriebs des Zählers ?

### Praktische Übung P5.2-3: Rechteckgenerator mit externer Taktung

Diese Übung kann nur mit dem Praktikumsrechner durchgeführt werden.

- I. Der Zähler #3 soll extern über den Eingang  $\overline{C3}$  ohne Frequenzteiler (1/8) getaktet werden. Welche Bits müssen Sie im Steuerregister CR#3 ändern ?
- II. Verbinden Sie den Ausgang CB2 des Portbausteins MC6821 mit dem Takteingang  $\overline{C3}$  des Bausteins MC6840. Stellen Sie mit Hilfe des Programms „Rechteckgenerator“ aus Übung P4.2-3, Kapitel 4, am Ausgang CB2 eine Triggerfrequenz von ca. 100 Hz ein. Laden Sie den Wert \$0100 in das Auffangregister des Zählers #3. Berechnen und messen Sie die Puls- und Pausendauer des Ausgangs O3 !
- III. Schreiben Sie den Wert ins Auffangregister von Zähler #3, durch den eine Ausgangsschwingung an O3 von (ca.) 1 Hz erzeugt wird !
- IV. Schätzen Sie die maximale Frequenz ab, die man durch die vorgegebenen Versuchsbedingungen erzeugen kann.
- V. Funktioniert die Schaltung auch mit  $\overline{G3} = H$  ?

<sup>1</sup> Es reicht, die geeigneten Werte direkt in die Register einzuschreiben, also ohne Programmerstellung.

**Praktische Übung P5.2-4: Monoflop**

I. Mit welchem Hexadezimalwert muß das Steuerregister CR#3 geladen werden, um folgende Betriebsart zu erhalten:

1. Ausgang O3 aktiviert,
2. Interrupt zur CPU gesperrt,
3. Single-Shot-Betrieb,
4. 16-bit-Zählmodus,
5. Takt intern vom Systemtakt E abgeleitet (Frequenz 1MHz),
6. Takt wird auf 1/8 heruntergeteilt.

Im Single-Shot-Betrieb ist das Ausgangssignal O3 nicht periodisch. Es kann daher nicht besonders gut auf der Erweiterungskarte beobachtet werden.

Abhilfe: Das Monoflop wird in periodischen Zeitabständen getriggert. Beim MC6840 kann dies auf zwei Arten geschehen.

- Durch die Software: Durch Setzen von Bit 0 im Steuerregister CR#1 wird der Zähler gestoppt, durch Rücksetzen gestartet, d.h. durch zyklisches Pulsen dieses Bits entsteht am Ausgang O3 eine periodische Schwingung.
- Durch die Hardware: Stoppen durch ansteigende Flanke am  $\overline{G3}$  Eingang, Starten durch fallende Flanke.

II. Starten durch die Software: (Praktikumsrechner oder 6809-Simulator)

Schreiben Sie ein Programm, das den Zähler #3 periodisch startet. Die Monoflop-Impulsdauer soll maximale Länge haben. Die periodische Rechteck-Schwingung zur Triggerung des Zählers soll über CB2 ausgegeben werden, ihre Schwingungsdauer über die Tastatur eingegbar sein.

III. Starten durch die Hardware: (nur mit dem Praktikumsrechner)

Jetzt soll der Zähler #3 intern getaktet, aber extern über den Eingang  $\overline{G3}$  getriggert werden. Ändern Sie das Programm aus II. so ab, daß über den Ausgang CB2 der Zähler #3 periodisch getriggert wird. Geben Sie eine Schwingungsdauer von ca. 1 s ein. Skizzieren Sie die Ausgangssignale an CB2 und O3.



### 5.2.2 Zykluszeit-/Impulsdauermessung

Wie bereits gesagt, kann man mit dem Zeitgeber-/Zählerbaustein nicht nur Signale erzeugen („Wave Synthesis Mode“), sondern auch messen („Wave Measurement Mode“). In realen Anwendungen kommen diese auszumessenden Signale von der Peripherie eines Mikrorechners, z.B. von einem Meßgerät. Da wir Ihnen einerseits solch ein Gerät nicht zur Verfügung stellen können, andererseits der 6809-Simulator die Nachbildung von Signalen am Steuereingang  $\overline{G3}$  des Zählers nicht unterstützt, müssen wir in den folgenden Praktischen Übungen diese Signale vom Mikroprozessor selbst erzeugen lassen und auf den Baustein MC6840 zurückkoppeln. Diese Übungen sind nur mit dem Praktikumsrechner auszuführen.

#### Praktische Übung 5.2-5: Messung von Impulsbreite und Schwingungsdauer

- I. Schreiben Sie ein Programm, das die Länge der negativen Impulsdauer einer Schwingung am Steuereingang  $\overline{G3}$  des Zählers #3 mißt.  
Gehen Sie dazu in folgenden Schritten vor:
  - Laden Sie das Steuerregister von Zähler #3 so, daß
    1. der Ausgang O3 aktiviert ist,
    2. Interrupts zur CPU zugelassen sind,
    3. der geeignete Modus zum Impulsbreiten-Vergleich gewählt wird,
    4. im 16-bit-Modus gezählt wird,
    5. der interne Takt E zum Zählen benutzt wird,
    6. dieser Takt ungeteilt verwendet wird,
    7. der Zählzyklus maximale Länge hat.
  - Ändern Sie den Interruptvektor des Zählers #3 so, daß eine eigenerstellte Interrupt-routine ausgeführt wird.
  - Erweitern Sie das Programm um eine Endlosschleife, in der ein symmetrisches Rechtecksignal über die Leitung CB2 des Portbausteins MC6821 ausgegeben wird. Der Wert, der die Länge der Impulspause und Impulsdauer in ms vorgibt, soll über die Tastatur eingegeben und in der Anzeige S7-S4 angezeigt werden. Achtung: während der Eingabe darf kein Interrupt von Zähler #3 „durchkommen“.
  - Schreiben Sie eine Interruptroutine zu Zähler #3, in der die gemessene Impulslänge in der Anzeige S3-S0 dargestellt wird. Achtung: Löschen des Interrupt-Flags nicht vergessen.
- II. Ändern Sie das Programm aus I. nun so ab, daß es unter den gleichen Bedingungen die Schwingungsdauer des Signals an CB2 mißt und anzeigt.
- III. Geben Sie in einer Tabelle die gemessenen Werte für Impulspause und Schwingungsdauer an, wenn der Zeitparameter für die Endlosschleife die Werte \$1, \$2, \$10, \$80 annimmt.
- IV. Was beobachten Sie an den Ausgängen CB2 und O3, wenn der Zeitwert \$1000 gewählt wird ?

**Praktische Übung 5.2-6: Messen von Schwingungen niedriger Frequenz**

Aufgabe P5.2-5, IV. zeigte, daß mit dem dort benutzten Programm nur Schwingungen ausgemessen werden konnten, bei denen die Impulsdauer oder Schwingungsdauer kürzer ist als der maximale Zählzyklus des Zählers #3 (Initialwert \$FFFF). Dieser Zyklus ist ca. 65 ms lang, falls der Frequenzteiler 1:8 nicht benutzt wird, und ca. 0.52 s mit Frequenzteiler.

Erweitern Sie nun das Programm aus P5.2-5 so, daß diese Restriktion aufgehoben wird. Gehen Sie dazu in folgenden Schritten vor:

1. Ändern Sie die Programmierung des Zählers #3 so, daß jeder Ablauf des Zählers zu einem Interrupt führt. In der Interruptroutine zählen Sie nun die Anzahl N der vollständig abgearbeiteten Zählzyklen.
2. Ergänzen Sie das Programm so, daß zum Ende jeder Vollschrwingung am Ausgang CB2 ein Interrupt über den Eingang CB1 des MC6821 erzeugt wird. Geben Sie in der Interruptroutine den unter 1. ermittelten Zählerstand in den Anzeigestellen S7-S4 aus und stellen Sie die abgelaufene Dauer D des aktuellen Zählzyklus in den Stellen S3-S0 dar.
3. Für die Gesamtzeit T der gemessenen Impulsdauer oder Schwingungsdauer ergibt sich damit:

$$T = N * \$10000 + D$$

4. Das Programm soll die Schritte 1. und 2. zyklisch wiederholen. Achtung: Löschen der Zähler und der Interrupt-Flags nicht vergessen !

## 5.3 Praktische Übungen zum Praktikumsrechner

Nachdem wir in den vorausgehenden Kapiteln und Abschnitten alle Komponenten des Praktikumsrechners ausführlich beschrieben haben, wollen wir Ihnen nun in diesem letzten Abschnitt Gelegenheit geben, sich in Praktischen Übungen mit allen diesen Komponenten noch einmal im Zusammenhang zu beschäftigen. Am Ende des Kapitels finden Sie zu jeder Übung eine Musterlösung. Bewußt haben wir bei diesen Lösungen auf die Angabe von Kommentaren verzichtet. Denn Sie sollen nur im "Notfall", also dann, wenn Sie keine eigene Lösung finden, darauf zurückgreifen. In diesem Fall sollen Sie den Programmablauf selbst analysieren und herausfinden, was durch die einzelnen Befehle geleistet wird. (In der Kopfzeile der Aufgaben sind jeweils die Komponenten angegeben, die in den Aufgaben besonders behandelt werden.)

### Praktische Übung P5.3-1: (Komponente: CPU)

*Es soll ein Programm geschrieben werden, das einen vierstelligen Dezimal-/Hexadezimal-Zähler realisiert. Der Zählmodus, die Schrittweite des Zählers sowie der Anfangswert des Zählzyklus sollen über die Tastatur einzugeben sein.*

Im einzelnen soll das Programm


1. die Anzeige löschen.
2. mit dem Startwert '0000' beginnend, ca. im 1/4-Sekunden-Takt mit der Schrittweite 1 in den Anzeigestellen S5-S2 (Adreßfeld) aufwärtszählen.
3. dabei die Tastatur abfragen und folgendermaßen auf die Tasten reagieren:
  - Taste "R": Rücksetzen des Zählerzustandes auf '0000', Fortsetzen des Zählvorganges.
  - Taste "S": Anhalten des Zählvorganges, solange die Taste gedrückt wird.
  - Taste "A": Eingabe eines vierstelligen Anfangswertes (z.B. mit SHOWADR, beendet durch "+"), danach Fortsetzen des Zählvorganges.
  - Taste "+": Wahl des hexadezimalen Zählmodus.
  - Taste "-": Wahl des dezimalen Zählmodus.
  - "0,...,F": Datentasten: Vorgabe der Zählerschrittweite 0–15. Diese bleibt bis zur nächsten Eingabe unverändert.

Alle anderen Funktionstasten sollen ohne Wirkung sein.

### Praktische Übung P5.3-2: (Komponente: CPU)

Gesucht ist ein Programm, das eine eingegebene Dezimalzahl  $Z=d_2d_1d_0$  (mit  $0 \leq Z \leq 127$ ) in die entsprechende Hexadezimalzahl umwandelt. Die Darstellung soll wahlweise im Zweierkomplement erfolgen.

Im einzelnen soll Ihr Programm

1. die Anzeige löschen,
2. sukzessiv die drei Dezimalziffern  $d_2, d_1, d_0$  einlesen und sofort in den Anzeigestellen  $S_4, S_3, S_2$  des Adreßfeldes darstellen; dabei sollen alle anderen Tasten (A,...,F und Funktionstasten) nicht akzeptiert werden,
3. feststellen, ob die eingegebene Zahl im vorgegebenen Bereich  $0 \leq Z \leq 127$  liegt;
  - ist dies nicht der Fall, so soll nach ca. 1 Sekunde das Programm von vorne begonnen werden (Punkt 1.),
  - liegt sie im Bereich, muß sie in die Hexadezimalform umgerechnet werden (s. Hinweis),
4. danach die Tastatur abfragen und nach Drücken der
  - **Taste 'S'** zur Eingabe der nächsten Zahl zum Anfang zurückspringen,
  - **Taste '+'** im Operationsfeld (Stelle  $S_7$ ) das stilisierte '+'-Zeichen "  " sowie im Datenfeld (Stellen  $S_1, S_0$ ) die Hexadezimalzahl darstellen,
  - **Taste '-'** im Operationsfeld (Stelle  $S_7$ ) das Zeichen '-' sowie im Datenfeld (Stellen  $S_1, S_0$ ) die Hexadezimalzahl im Zweierkomplement darstellen.

**Hinweis:**  $Z = d_2d_1d_0 = d_2 \cdot 100 + d_1 \cdot 10 + d_0 = d_2 \cdot \$64 + d_1 \cdot \$A + d_0$  .

**Praktische Übung P5.3-3: (Komponente: CPU)**

*Es soll ein Programm geschrieben werden, das die Eingabe einer vierstelligen Hexadezimalzahl und die wahlweise Darstellung dieser Zahl in "normaler" oder "invertierter" Form erlaubt. Unter "invertiert" soll dabei verstanden werden, daß die zunächst aktivierten LEDs der 7-Segment-Anzeige ausgeschaltet, die nicht aktivierten eingeschaltet werden.*

Im einzelnen soll das Programm,

1. die Anzeige löschen,
2. die Eingabe der vierstelligen Hexadezimal-Zahl vornehmen. Diese Eingabe soll von links nach rechts geschehen, und die eingegebenen Ziffern sollen in den Anzeigestellen S5-S2 (Adreßfeld der Anzeige) dargestellt werden.
3. die Tastatur zyklisch abfragen und folgendermaßen auf die Tasten reagieren:
  - Taste "+": Darstellung der Zahl (in S5-S2) in normaler Form, also so wie sie eingegeben wurde.
  - Taste "-": Darstellung der Zahl (in S5-S2) in invertierter Form.
  - Taste "R": - automatischer Wechsel zwischen beiden Darstellungsformen ca. im 1/4-Sekunden-Takt. Durch Betätigen der
    - Funktionstaste "A": Abbruch des Wechsels in der Darstellung und Rücksprung zu 3.
  - Taste "S": Verzweigung zu 1. zur Eingabe einer neuen Zahl.

**Hinweise:**

Alle anderen Funktionstasten brauchen nicht berücksichtigt zu werden. Zur Ziffern-Darstellung in invertierter Form können Sie die folgende Tabelle verwenden, die für jede Hex.-Ziffer 0,...,F die erforderliche Codierung zur Ansteuerung der Segmente in hexadezimaler Form angibt.

Hex.-Ziffer	0	1	2	3	4	5	6	7
Codierung	\$40	\$79	\$24	\$30	\$19	\$12	\$02	\$78
Hex.-Ziffer	8	9	A	B	C	D	E	F
Codierung	\$00	\$18	\$08	\$03	\$27	\$21	\$06	\$0E

**Praktische Übung P5.3-4: (Komponente: CPU)**

*Es soll ein Programm geschrieben werden, das eine Folge von 4-stelligen Hexadezimalzahlen einliest, in der Anzeige darstellt und in einem Benutzer-Stack ablegt. Gleichzeitig soll die Nummer des jeweiligen Stack-Eintrags dargestellt werden. Die Übernahme der Zahl in den Stack wird durch die Taste '+' ausgelöst. Durch die Taste '-' kann der aktuelle Stack-Inhalt in umgekehrter Reihenfolge abgerufen werden. Dabei soll eine Überprüfung auf "Stack Underflow" stattfinden und das Programm ggfs. von vorne beginnen.*

Im einzelnen soll das Programm,

1. die Anzeige löschen,
2. eine 4-stellige Hexadezimal-Zahl als Initialwert für den Benutzer-Stackpointer einlesen und in den Anzeigestellen S5,...,S2 darstellen; (Die Eingabe soll durch die Funktionstaste 'Ad' ausgelöst werden, bei allen anderen Funktionstasten Rücksprung zu Punkt 1.)
3. die Nummer des aktuellen Stackeintrags in den Anzeigestellen S1, S0 ausgeben;
4. die Tastatur zyklisch abfragen, eine 4-stellige Hexadezimalzahl einlesen und in den Anzeigestellen S5,...,S2 darstellen;
5. nach Betätigen der **Taste '+'**:  
die Zahl in den Benutzer-Stack übertragen, den Zähler für den Stackeintrag um '1' erhöhen und mit Punkt 3. fortfahren;
6. nach Betätigen der **Taste '-'**:  
den Zähler für den Stackeintrag um '1' erniedrigen
  - falls Zählerwert = 0:
    - die Anzeige löschen,
    - **'SU'** (*Stack Underflow*) in S7, S6 für ca. 1 Sekunde ausgeben,
    - mit Punkt 1. fortfahren;
  - falls Zählerwert > 0:  
selektierten Stackeintrag in S5,...,S2 darstellen;
7. die Tastatur abfragen und nach Drücken der
  - **Taste '-'**: mit Punkt 6. fortfahren,
  - aller anderen Tasten: mit Punkt 3. fortfahren.

**Praktische Übung P5.3-5: (Komponente: CPU)**

*Es soll ein Programm geschrieben werden, das eine verschlüsselte Ausgabe des Speicherinhalts ermöglicht. Dazu soll zunächst als Code eine Permutation der Ziffern {0,...,9,A,..F} eingegeben werden und danach jedes auszugebende Datum mit diesem Code umgesetzt werden.*

Im einzelnen soll das Programm,

1. die Anzeige löschen,
2. den benutzten Code als Permutation der Hex-Ziffern 0,...,F einlesen. Dazu soll sukzessiv ca. im Halbsekundentakt die jeweilige Ziffer in der Anzeigestelle S1 ausgegeben, die Stelle S0 gelöscht und danach über die Tastatur der Codewert eingelesen und in S0 dargestellt werden.
3. danach sequentiell ca. im Halbsekundentakt den Speicherbereich ab Adresse \$0400 lesen und verschlüsselt in der Anzeige in der folgenden Form ausgeben:
  - Anzeigestellen S5-S2: Adresse der aktuelle dargestellten Speicherzelle
  - Anzeigestellen S1-S0: Wert der adressierten Speicherzelle
  - Anzeigestellen S7-S6: Wert der Speicherzelle, aber umgewandelt mit dem zunächst eingegebenen Code.
4. die unter 3. beschriebene Darstellung des Speicherbereichs mit einem Sprung ins Monitorprogramm abbrechen, sobald die Funktionstaste "S" betätigt wird.

**Hinweis:**

Die korrekte Eingabe des Codes unter 2. als Permutation der Hex-Ziffern obliegt dem Benutzer und muß nicht durch das Programm überprüft werden.

**Praktische Übung P5.3-6: (Komponente: CPU)**

*Es soll ein Programm geschrieben werden, das eine eingegebene zweistellige Hex-Zahl verschlüsselt wieder ausgibt. Dazu soll zunächst als Code eine Permutation der Ziffern {0,...,9,A,..F} eingegeben werden und danach jedes eingelesene Datum mit diesem Code umgesetzt werden.*

Im einzelnen soll das Programm,

1. den benutzten Code als Permutation der Hex-Ziffern 0,...,F einlesen. Dazu sollen sukzessiv jeweils 4 Hex-Ziffern (0123/4567/89AB/CDEF) in den Anzeigestellen S7-S4 ausgegeben und die zugehörigen Codewerte eingelesen und in den Stellen S3-S0 dargestellt werden.
2. danach
  - die Anzeige löschen,
  - eine zweistellige Hex-Zahl einlesen und in den Stellen S1,S0 darstellen,
  - die eingelesene Zahl ziffernweise mit dem unter 2. vorgegebenen Code umwandeln und in den Anzeigestellen S7,S6 ausgeben
3. nach einer Verzögerung von ca. einer halben Sekunde mit Punkt 2. fortfahren.

**Hinweise:**

Zur Eingabe des Codes bzw. der umzuwandelnden Daten benutzen Sie am besten die Routinen SHOWADR und SHOWDATA. Die korrekte Eingabe des Codes unter 2. als Permutation der Hex-Ziffern obliegt dem Benutzer und muß nicht durch das Programm überprüft werden.



**Praktische Übung P5.3-7: (Komponente: Anzeige)**

*Es soll ein Programm geschrieben werden, das eine Folge von Hexadezimal-Ziffern einliest und in einem Benutzer-Stack ablegt. Dabei soll jedes eingegebene Zeichen in der Anzeige in einer Art Balkendarstellung ausgegeben werden. Außerdem soll der Stackinhalt in beiden Richtungen fortlaufend in der Anzeige dargestellt werden können.*

Im einzelnen soll das Programm,

1. die Anzeige löschen;
2. danach zyklisch die Tastatur abfragen und beim Betätigen
  - einer **Datentaste k** aus  $\{0, \dots, 9, A, \dots, F\}$ :
    - k in einem Benutzer-Stack ab Adresse \$1000 abspeichern,
    - k als Balkendarstellung durch das Zeichen '-' (Segment g) in allen Anzeigestellen  $S_j$  mit  $0 \leq j \leq (k \bmod 8)$  ausgeben;
  - der **Funktionstaste '-'**:  
sukzessiv alle bereits eingelesenen Stackwerte in umgekehrter Eingabereihenfolge ca. im Halbsekunden-Takt und in der eben beschriebenen Balkendarstellung ausgeben;
  - der **Funktionstaste '+'**:  
sukzessiv alle bereits eingelesenen Stackwerte in der Eingabereihenfolge ca. im Halbsekunden-Takt und in der eben beschriebenen Balkendarstellung ausgeben.

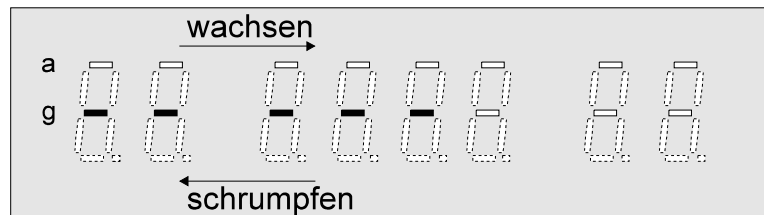
**Hinweise:**

1. Die Tasten '+' und '-' sollen mit gleicher Funktion beliebig oft zu betätigen sein.
2. Alle nicht genannten Funktionstasten sollen keine Funktion haben.
3. Die Darstellung der eingegebenen Daten als Balken geschieht am besten in einem Unterprogramm.
4. Die sukzessive Darstellung der Stackwerte darf den aktuellen Stackbereich nicht überschreiten.

### Praktische Übung P5.3-8: (Komponente: Anzeige)

Es ist ein Programm zu schreiben, daß eine waagerechte "Balkenanzeige" (in Form einer Segmentreihe) in der Anzeige nachbildet. Dabei sollen wahlweise das oberste oder das mittlere Segment jeder Anzeigestelle benutzt werden können. Ausgehend von der "gelöschten" Anzeige, soll der "Balken" solange nach rechts länger bzw. nach links kürzer werden, wie bestimmte Tasten gedrückt werden. In jedem Zustand soll ein Rücksprung in den Anfangszustand zur Auswahl einer Segmentreihe möglich sein.

Skizze:



Im einzelnen soll das Programm:

1. zunächst die Anzeige löschen,
2. die Auswahl der Segmentreihe (des Balkens) durch die Kennung 'E' in der Anzeigestelle S0 anfordern:
  - Funktionstaste 'A': oberste Segmentreihe (Segment a),
  - Funktionstaste 'G': mittlere Segmentreihe (Segment g),
 und danach zyklisch die Tastatur abfragen und nach Betätigen der
3. Taste '+': den "Balken", wie oben beschrieben, (ungefähr) im Halb-Sekundentakt solange nach rechts wachsen lassen, wie die Taste '+' gedrückt ist, und bei Erreichen der Anzeigestelle S0 stoppen,
4. Taste '-': den "Balken" (ungefähr) im Halb-Sekundentakt solange nach links schrumpfen lassen, wie die Taste '-' gedrückt ist, und nach Löschen der Anzeigestelle S7 stoppen,
5. Taste 'R': mit der Auswahl einer neuen Segmentreihe unter 2. fortfahren.

(Alle Datentasten und nicht erwähnten Funktionstasten sind ohne Funktion)

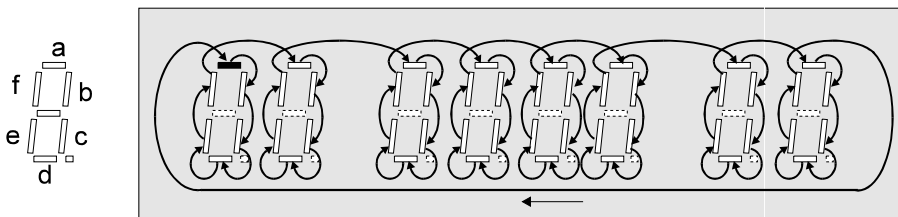
#### Hinweis:

Die Benutzung der Routinen zur Bearbeitung des Anzeigepuffers (CLDBUF, SHOWDBUF, RRDBUF, RLDBUF, COPYDBUF) erleichtert die Lösung der Aufgabe.

**Praktische Übung P5.3-9: (Komponente: Anzeige)**

Es ist ein Programm zu schreiben, daß ein "Lauchlicht" (in Form eines einzelnen Leuchtpunktes) zyklisch nach rechts durch die Anzeige rotiert (, so daß der rechts "herausfallende" Leuchtpunkt links wieder erscheint). Bei dieser Rotation sollen die Rand-segmente (a,b,c,d,e,f; s. Skizze) der Anzeigestellen im Uhrzeigersinn durchlaufen werden. Die Rotation soll wahlweise segmentweise auf Anforderung oder aber kontinuierlich geschehen. In jedem Zustand soll ein Rücksetzen in den Anfangszustand möglich sein.

Skizze:



Im einzelnen soll das Programm:

1. zunächst die Anzeige löschen,
  2. den Anfangszustand herstellen, in dem nur das Segment a der Anzeigestelle S7 aktiviert ist,
- und danach zyklisch die Tastatur abfragen und nach Betätigen der
3. Taste '+': den "Leuchtpunkt" kontinuierlich, wie oben beschrieben, (ungefähr) im Viertel-Sekundentakt nach rechts durch die Anzeige rotieren lassen,  
Taste '-': die kontinuierliche Rotation stoppen,
  4. Taste 'S': den "Leuchtpunkt", wie oben beschrieben, (ungefähr) im Viertel-Sekundentakt solange nach rechts durch die Anzeige rotieren lassen, wie die Taste 'S' gedrückt ist,
  5. Taste 'R': den unter 2. beschriebenen Anfangszustand wieder herstellt.

(Alle Datentasten und nicht erwähnten Funktionstasten sind ohne Funktion)

**Hinweis:**

Die Benutzung der Routinen zur Bearbeitung des Anzeigepuffers (CLDBUF, SHOWDBUF, RRDBUF, RLDBUF, COPYDBUF) erleichtert die Lösung der Aufgabe.

**Praktische Übung P5.3-10: (Komponente: Tastatur)**

*Es soll ein Programm geschrieben werden, das bei jedem Tastendruck die Zeile bzw. Spalte der Tastatur anzeigt, in der die gedrückte Taste liegt. Außerdem sollen der gewählte Modus "Zeile/Spalte" sowie der gedrückte Tastenwert dargestellt werden.*

Im einzelnen soll das Programm,

1. die Anzeige löschen und die Voreinstellung des "Zeilen-Modus" durch ein "L" in der Anzeigestelle S7 ausgeben;
2. danach zyklisch die Tastatur abfragen und beim Betätigen
  - der **Funktionstaste 'L'**:  
die Anzeige löschen, in den "Zeilen-Modus" schalten und dabei ein "L" in der Anzeigestelle S7 ausgeben;
  - der **Funktionstaste 'S'**:  
die Anzeige löschen, in den "Spalten-Modus" schalten und dabei ein "S" in der Anzeigestelle S7 ausgeben;
  - einer **Datentaste k** aus {0,...,9,A,...,F}:
    - den Wert k in der Anzeigestelle S0 ausgeben;
    - im "Zeilen-Modus" die Zeile der Datentastatur in den Anzeigestellen S5-S2 ausgeben, in der k liegt;
    - im "Spalten-Modus" die Spalte der Datentastatur in den Anzeigestellen S5-S2 ausgeben, in der k liegt.

Alle anderen Funktionstasten (außer "L", "S") sollen keine Funktion haben.

**Beispiel:**

<u>Taste</u>	<u>Anzeige</u>	
L	L_ _ _ _ _	__ gelöschte Anzeigestelle
5	L_ 45 67 _5	
E	L_ CDEF _E	
S	S_ _ _ _ _	
5	S_ 159D _5	

**Praktische Übung P5.3-11: (Komponente: Tastatur)**

*Es soll ein Programm geschrieben werden, das bei jedem Druck einer Datentaste die Position der betätigten Taste anzeigt. Die Darstellung soll wahlweise in zwei verschiedenen Modi möglich sein: durch Angabe der Zeilen- und Spaltennummer oder des "Quadranten", in dem die gedrückte Taste liegt. Außerdem sollen der gewählte Modus "Zeile&Spalte" bzw. "Quadrant" sowie der gedrückte Tastenwert dargestellt werden.*

Im einzelnen soll das Programm,

1. die Anzeige löschen und die Voreinstellung des "Zeilen&Spalten"-Modus durch eine "0" in der Anzeigestelle S7 anzeigen;
2. danach zyklisch die Tastatur abfragen und beim Betätigen
  - der **Funktionstaste 'A'**:  
die Anzeige löschen, in den "Zeilen&Spalten"-Modus schalten und dabei ein "0" in der Anzeigestelle S7 ausgeben;
  - der **Funktionstaste 'D'**:  
die Anzeige löschen, in den "Quadranten"-Modus schalten und dabei eine "1" in der Anzeigestelle S7 ausgeben;
  - einer **Datentaste k** aus {0,...,9,A,...,F}:
    - den Wert k in der Anzeigestelle S0 ausgeben;
    - **im "Zeilen&Spalten"-Modus**: in der Anzeigestelle S3 die Zeile 0-3 der Datentastatur und in S2 die Spalte 0-3 ausgeben, in der k liegt;
    - **im "Quadranten"-Modus**: in der Anzeigestelle S3 angeben, ob die Taste in der oberen (Ausgabe "o") oder unteren Hälfte (Ausgabe "u") liegt, in der Anzeigestelle S2, ob die Taste in der linken (Ausgabe "l") oder rechten Hälfte (Ausgabe "r") liegt.

Alle anderen Funktionstasten (außer "A", "D") sollen keine Funktion haben. Die Nummerierung der Zeilen und Spalten sei so gewählt, daß die Datentaste "0" in der 0. Zeile und 0. Spalte, die Datentaste "F" in der 3. Zeile und 3. Spalte liegt.

**Beispiel:**      Taste      Anzeige

A	0_ _ _ _ _
9	0_ _21 _9
3	0_ _03 _3
D	1_ _ _ _ _
9	1_ _o l _9
3	1_ _u r _3

\_ gelöschte Anzeigestelle

**Praktische Übung P5.3-12: (Komponente: Tastatur)**

*Es soll ein Programm geschrieben werden, das eine Folge von Tasteneingaben entgegen nimmt und die Gruppenzugehörigkeit (Datentaste oder Funktionstaste) jeder betätigten Taste in bestimmter Form in der Anzeige ausgibt. Sobald eine vorgegebene Obergrenze für eine der Tastengruppen erreicht ist, sollen die Anzahlen der betätigten Tasten jeder Gruppe ausgegeben werden und das Programm von vorne beginnen.*

Im einzelnen soll das Programm,

1. die Anzeige löschen,
2. auf den Start durch die Funktionstaste 'S' warten und danach den Ablauf um ca. eine halbe Sekunde verzögern, (alle anderen Tasten sollen hier keine Funktion haben.)
3. danach zyklisch die Tastatur abfragen und beim Betätigen
  - einer **Datentaste k** aus  $\{0, \dots, 9, A, \dots, F\}$ :
    - einen (internen) Zähler  $Z_D$  für die Betätigung einer Datentaste um '1' erhöhen,
    - in der Anzeigestelle  $S_j$  mit  $j = k \bmod 8$  das Zeichen 'd' solange ausgeben, wie die Taste gedrückt bleibt;
  - einer **Funktionstaste** aus  $\{+, -, Ad, Da, \dots, F3\}$ :
    - einen (internen) Zähler  $Z_F$  für die Betätigung einer Funktionstaste um '1' erhöhen,
    - für den internen Tastencode  $\$8X$  ( $X=0, \dots, B$ ) in der Anzeigestelle  $S_j$  mit  $j = X \bmod 8$  das Zeichen 'F' solange ausgeben, wie die Taste gedrückt bleibt;
4. Sobald einer der internen Zähler  $Z_D$  oder  $Z_F$  den Hex.-Wert  $\$A$  erreicht, sollen beide Zählerstände in der Anzeige dargestellt werden ( $Z_D$  in  $S_7$ ,  $Z_F$  in  $S_0$ ) und danach mit Punkt 1. fortgefahren werden.

**Praktische Übung P5.3-13: (Komponente: Anzeige)**

*Es ist ein Programm zu schreiben, daß eine 8stellige Dezimalzahl zyklisch nach rechts durch die Anzeige rotiert (, so daß die rechts "herausfallende" Ziffer links wieder erscheint). Die Rotation soll wahlweise stellenweise auf Anforderung oder aber kontinuierlich geschehen. In jedem Zustand soll ein Rücksetzen in die Ausgangslage möglich sein.*

Im einzelnen soll das Programm:

1. die Anzeige löschen,
2. die 8stellige Dezimalzahl Z ziffernweise einlesen und gleichzeitig, mit der Anzeigestelle S7 beginnend, in der Anzeige darstellen; (dabei sollen alle Funktionstasten und die Datentasten 'A,B,C,D,E,F' unberücksichtigt bleiben)

und danach zyklisch die Tastatur abfragt und nach Betätigen der

3. Taste '+': die Zahl Z in der Anzeige um eine Stelle nach rechts rotiert,
4. Taste 'G': die Zahl Z kontinuierlich (ungefähr) im Sekundentakt nach rechts durch die Anzeige rotiert,  
Taste 'S': die kontinuierliche Rotation stoppt, (alle anderen Tasten ohne Funktion)
5. Taste 'R': die ursprünglich eingegebene Zahl Z wieder darstellt.

**Beispiel:**

Eingabe Z: 4 5 2 1 7 9 8 0

Taste '+': 0 4 5 2 1 7 9 8

Taste '+': 8 0 4 5 2 1 7 9

Taste 'G': 9 8 0 4 5 2 1 7

7 9 8 0 4 5 2 1

1 7 9 8 0 4 5 2 ....

Taste 'S': Stopp der Rotation

Taste 'R': 4 5 2 1 7 9 8 0

**Hinweis:**

Die Benutzung der Routinen zur Bearbeitung des Anzeigepuffers (CLDBUF, SHOWDBUF, RRDBUF, RLDBUF, COPYDBUF) erleichtert die Lösung der Aufgabe.

**Praktische Übung P5.3-14: (Monitorroutine)**

*Gesucht ist ein Programm, das den Offset eines Branch-Befehls in hexadezimaler Form berechnet. Dazu muß als Startadresse die Adresse des ersten OpCodes hinter dem Branch-Befehl und als Endadresse die Adresse des OpCodes angegeben werden, zu dem das Programm verzweigen soll. Bei Rückwärtssprüngen muß das Offset im Zweierkomplement angegeben werden.*

Im einzelnen soll das Programm,

1. die Anzeige löschen,
2. ins Operationsfeld der Anzeige (Stellen S6,S7) die Kennung "SA" ("Startadresse") schreiben und dann eine vierstellige Hexadezimaladresse einlesen und diese im Adreßfeld (S5-S2) darstellen,
3. ins Operationsfeld der Anzeige die Kennung "EA" ("Endadresse") schreiben und dann eine vierstellige Hexadezimaladresse einlesen und diese im Adreßfeld (S5-S2) darstellen,
4. feststellen, ob ein *short branch* oder ein *long branch* vorliegt und dies durch ein "L" bzw. "S" im Operationsfeld (Stelle S7) anzeigen; (ein *long branch* liegt vor, falls die Anzahl der zu überspringenden Bytes größer als 127 bzw. kleiner als -128 ist)
5. das errechnete hexadezimale Offset eines *short branch* in den Stellen S2,S3 bzw. eines *long branch* in den Stellen S5-S2 darstellen,
6. nach Drücken der **Taste '+'** zur Ausführung der nächsten Offset-Berechnung zum Anfang zurückspringen (Punkt 1.),
7. nach Drücken der **Taste '-'** die eingegebene Startadresse und Zieladresse nebeneinander in der Anzeige (S7-S0) darstellen.



**Praktische Übung P5.3-15: (Monitorroutine)**

*In dieser Aufgabe soll ein kleiner Speicherbereich nach dem Auftreten zweier einzugebener Zahlen untersucht und alle Speicherzellen "zwischen" den Fundorten angezeigt werden. Wird eine Zahl nicht gefunden, soll sie nach dem Durchsuchen dargestellt werden.*

Schreiben Sie ein Programm, daß

1. zunächst die Anzeige löscht,
2. in den Anzeigestellen S7,S6 die Kennung "E0" ("Eingabe 0") ausgibt und in den Stellen S5,S4 die zweistellige Hexadezimal-Zahl E0 einliest,
3. in den Anzeigestellen S3,S2 die Kennung "E1" ("Eingabe 1") ausgibt und in den Stellen S1,S0 die zweistellige Hexadezimal-Zahl E1 einliest,
4. die Anzeige löscht,
5. den Speicherbereich \$0400 - \$04FF sequentiell durchsucht und
6. falls E0 nicht gefunden wird: die Zahl E0 in S1,S0 ausgibt und das Programm beendet,
7. falls E0 gefunden wird:
  - mit der Adresse der Fundstelle beginnend, den Rest des Speicherbereichs sequentiell darstellt,
  - wobei in S5 - S2 die jeweilige Speicheradresse und in S1,S0 das zugehörige Datum ausgegeben werden soll und
  - zwischen den Darstellungen der Speicherzellen eine Zeitverzögerung von etwa einer halben Sekunde auftreten soll,
8. jedes ausgegebene Datum mit der Zahl E1 vergleicht und
9. falls E1 im untersuchten Speicherbereich \$0400 - \$04FF gefunden wird:
  - das Programm (mit der Darstellung der Fundadresse in S5 - S2 und der Zahl E1 in S1,S0) beendet,
10. falls E1 nicht gefunden wird: die Zahl E1 in S1,S0 ausgibt und das Programm beendet.

**Praktische Übung P5.3-16: (Monitorroutine)**

*In dieser Aufgabe sollen Sie einen kleinen Speicherbereich nach dem Auftreten zweier einzugebener Zahlen untersuchen und jeden "Fundort" einer der Zahlen kennzeichnen.*

Schreiben Sie ein Programm, das

1. zunächst die Anzeige löscht,
2. in den Anzeigestellen S7,S6 die Kennung "E0" ("Eingabe 0") ausgibt und in den Stellen S5,S4 die zweistellige Hexadezimal-Zahl E0 einliest,
3. in den Anzeigestellen S7,S6 die Kennung "E1" ("Eingabe 1") ausgibt und in den Stellen S5,S4 die zweistellige Hexadezimal-Zahl E1 einliest,
4. die Anzeige löscht,
5. den Speicherbereich \$0400 - \$04FF sequentiell durchsucht und in den Anzeigestellen S5 - S2 die jeweilige Speicheradresse, in S1,S0 das zugehörige Datum ausgibt, wobei zwischen den Darstellungen der Speicherzellen eine Zeitverzögerung von etwa einer halben Sekunde auftreten soll,
6. falls eine der eingegebenen Zahlen E0 oder E1 gefunden wird:  
gleichzeitig ihre Kennung ("E0", "E1") in den Stellen S7,S6 ausgibt und die Verzögerung der Darstellung auf zwei Sekunden verlängert,
7. auf Betätigen der Taste S ("Stop") den Suchvorgang jederzeit beendet. (Um die Zeitverzögerung zu erfassen, kann die Taste S dazu entsprechend lange gedrückt bleiben.)

**Praktische Übung P5.3-17: (Komponente: Parallel-Port PB)**

Schließen Sie zunächst die Schnittstellen-Erweiterungskarte am Praktikumsrechner an. Schreiben Sie nun ein Programm, das

1. die Anzeige löscht,
2. einen Rechteck-Takt mit einer Frequenz von 1 Hz erzeugt und den Zustand dieses Taktes im Datenfeld der Anzeige (Stelle  $S_0$ ) durch "0" bzw. "1" darstellt,
3. in Abhängigkeit von der Stellung der Schalter  $SW_0$  und  $SW_1$  an den Porteingängen  $PB_0, PB_1$  den erzeugten Takt auf einen der Ausgänge  $PB_7-PB_4$  gibt, wobei der Schaltzustand  $(SW_1, SW_0)$  [als Dezimalzahl aufgefaßt] + 4 gerade den Index  $i$  der Portleitung  $PB_i$  angeben soll,
4. durch die Tasten '+' bzw. '-' des Mikrorechners veranlaßt, den Ausgang CB2 so steuert, daß die angeschlossene LED aktiviert bzw. nicht aktiviert wird.

Die Abfrage der Tasten '+', '-' und der Schalter  $SW_0, SW_1$  braucht nur zu den Zeitpunkten der Zustandsänderungen des Taktes geschehen, so daß Sie zur Erzeugung der Taktzeiten die Routine DLY1MS benutzen können.

**Praktische Übung P5.3-18: (Komponente: Parallel-Port PB)**

*In dieser Aufgabe soll jede über die Tastatur eingegebene Hexadezimal-Ziffer in den ASCII-Code übersetzt und auf "Anforderung" auf dem Port PB ausgegeben werden. Die Ausgabe soll über CB2 angezeigt werden.*

Schließen Sie zunächst die Schnittstellen-Erweiterungskarte am Praktikumsrechner an. Schreiben Sie nun ein Programm, das

1. die Anzeige löscht,
2. den Port PB geeignet initialisiert,  
und danach zyklisch
3. die Datentasten der Tastatur abfragt, (die Funktionstasten sollen keine Funktion haben, auch nicht bei der Eingabe eines Datums!)
4. jedes eingegebene Datum (0,...,9,A,..F) in der Anzeigestelle S7 und seine ASCII-Darstellung in den Stellen S5,S4 ausgibt,
5. falls über den Schalter an CB1 keine Ausgabeanforderung gestellt wurde, mit 3. weitermacht,
6. falls über CB1 eine Ausgabeanforderung gestellt wurde:
  - die aktuelle Anforderung im Steuerregister löscht,
  - das zuletzt eingegebene Zeichen als ASCII-Zeichen auf den Port PB ausgibt und
  - diese Ausgabe durch die LED an CB2 und durch die Ziffer "0" in der Anzeigestelle S0 ca. eine Sekunde lang anzeigt.

<u>hex.</u>	<u>ASCII</u>	<u>hex.</u>	<u>ASCII</u>	<u>hex.</u>	<u>ASCII</u>	<u>hex.</u>	<u>ASCII</u>
0	30	4	34	8	38	C	43
1	31	5	35	9	39	D	44
2	32	6	36	A	41	E	45
3	33	7	37	B	42	F	46

**Praktische Übung P5.3-19: (Komponente: Parallel-Port PB)**

*In dieser Aufgabe sollen Sie die am Port PB anliegenden Binär-Informationen genau dann einlesen und in der Anzeige hexadezimal darstellen, wenn das Programm im "Einlesemodus" ist. Die Umschaltung zwischen den Modi "Einlesen/nicht Einlesen" soll durch die Tastatur geschehen.*

Schließen Sie zunächst die Schnittstellen-Erweiterungskarte am Praktikumsrechner an. Schreiben Sie nun ein Programm, das

1. die Anzeige löscht,
2. den Port PB geeignet initialisiert,

und danach zyklisch

3. die Tastatur auf Betätigung der Funktionstasten "+" und "-" abfragt, (alle anderen Tasten sollen keine Funktion haben!)
4. sich nach Betätigen der Taste "-":
  - im Modus 0 befindet, in dem keine Eingabe über den Port PB möglich ist, und
  - dies durch die Ziffer "0" in der Anzeigestelle S0 (alle anderen Anzeigestellen gelöscht !)
  - sowie durch eine ausgeschaltete LED an CB2 anzeigt,
  - Einlese-Anforderungen über den Schalter an CB1 ignoriert (und im Statusregister des Ports löscht),
5. sich nach Betätigen der Taste "+":
  - im Modus 1 befindet, in dem eine Eingabe über den Port PB möglich ist, und
  - dies durch die Ziffer "1" in der Anzeigestelle S0
  - sowie durch eine eingeschaltete LED an CB2 anzeigt,
  - Einlese-Anforderungen über den Schalter an CB1 auswertet, danach löscht und
  - die eingelesene Binär-Information als Hexadezimal-Zahl in den Stellen S7,S6 darstellt.

**Praktische Übung P5.3-20: (Komponente: Zeitgeber/Zähler)**

*In dieser Aufgabe sollen Sie mit Hilfe des Zeitgeber-/Zählerbausteins einen sog. Watch-Dog Timer (WDT) realisieren. Dieser soll während der Ausführung eines Programms überwachen, ob jeweils vor Ablauf eines Zeitintervalls eine gewünschte Reaktion der Rechnerperipherie stattfindet. Ist dies nicht der Fall, soll der WDT eine Fehlermeldung zum Anwender und zur Peripherie ausgeben.*

Im einzelnen soll das Hauptprogramm,

1. die Anzeige löschen,
2. den Zähler #3 des MC6840 so initialisieren (noch nicht starten!), daß er einen einzelnen Impuls maximaler Länge ausgeben und den Prozessor am Ende des Impulses durch einen Interrupt informieren kann,
3. den Anfangswert '0000' in den Anzeigestellen S5 - S2 ausgeben und auf die Betätigung der Tastatur warten,
4. nach Betätigen der Funktionstaste 'S' (für 'START'):
  - a) die Anzeigestellen S5 - S2 als 4-stelligen Hexadezimalzähler (ohne Zeitverzögerung) hochzählen,
  - b) den Zähler #3 des MC6840 starten,
5. bei jeder Betätigung der Funktionstaste 'R' (für 'RESET') den Zähler #3 des MC6840 auf den Anfangswert 'FFFF' zurücksetzen und erneut starten,

In der Interruptroutine zum Zähler #3 soll immer dann, wenn der Zähler #3 auf den Wert '0000' heruntergezählt wurde, also nicht rechtzeitig die Taste 'R' betätigt wurde:

1. der Zähler #3 gestoppt werden,
2. als Fehlermeldung in der Anzeigestelle S2 (nur) der Dezimalpunkt aktiviert werden,
3. auf die erneute Betätigung der Funktionstaste 'S' (für 'START') gewartet werden,
4. danach der Zähler #3 erneut mit dem Wert 'FFFF' gestartet
5. und der Zählvorgang des o.g. Hexadezimalzählers in den Anzeigestellen S5 - S2 ohne Zurücksetzen fortgesetzt werden.

**Hinweise:**

- Vergessen Sie nicht, das Interrupt Flag des Zählers #3 vor Verlassen der Interruptroutine zurückzusetzen.
- Alle jeweils nicht genannten Funktionstasten sollen keine Wirkung haben.

**Praktische Übung P5.3-21: (Komponente: Zeitgeber/Zähler)**

*In dieser Aufgabe soll ein vom Zeitgeber-/Zählerbaustein ausgegebenes Rechtecksignal durch den Prozessor überwacht werden. Sobald dieses Signal durch den Steuereingang des Zählers #3 zu lange unterbrochen wird, wird die Ausgabe gestoppt, bis sie eventuell vom Benutzer erneut gestartet wird.*

Im einzelnen soll das Hauptprogramm,

1. die Anzeige löschen,
2. den Zähler #3 des MC6840 so initialisieren (noch nicht starten!), daß er ein periodisches Rechtecksignal maximaler Dauer (minimaler Frequenz) ausgeben und den Prozessor am Ende jedes Zählzyklus durch einen Interrupt informieren kann,
3. den Anfangswert '0000' in den Anzeigestellen S5 - S2 und die Kennung 'S' (für 'STOP') in der Stelle S0 ausgeben und auf die Betätigung der Tastatur warten,
4. nach Betätigen der Funktionstaste 'G' (für 'GO'):
  - a) den Zähler #3 des MC6840 starten,
  - b) in der Anzeigestelle S0 die Kennung 'C' (für 'COUNT') ausgeben,
5. eine Zeitverzögerung von ca. 16 Sekunden ablaufen lassen,
6. feststellen, ob während der gesamten Verzögerungszeit der Zähler #3 über seinen Steuereingang G3 gesperrt war und
  - den Zähler #3 stoppen und zu 3. zurückspringen, falls dies der Fall war,
  - mit 5. weitermachen, falls es nicht der Fall war.

In der Interruptroutine des Zählers #3 soll

1. der Hexadezimal-Zähler in den Anzeigestellen S5 - S2 nach jedem Zählzyklus des Zählers #3 um 1 erhöht werden,
2. das Hauptprogramm geeignet darüber informiert werden, daß der Zähler #3 einen Zählzyklus beendet hat und also nicht gesperrt war.

**Hinweise:**

- Vergessen Sie nicht, das Interrupt Flag des Zählers #3 vor Verlassen der Interruptroutine zurückzusetzen.
- Alle jeweils nicht genannten Funktionstasten sollen keine Wirkung haben.





## Lösungsvorschläge zu den Praktischen Übungen

### Zu P5.2-1:

Beachten Sie bitte, daß Sie die in die Steuerregister eingegebenen Werte nicht im Datenfeld der Anzeige lesen können, da es sich um Nur-Schreib-Register handelt.

- I. Zum Ansprechen des Steuerregisters von Zähler #3 muß zunächst das Bit 0 im Steuerregister CR#2 auf 0 gesetzt werden:

Taste A:      \$F019   (\$00)

Taste D:              \$00

### „Programmieren“ des Zählers #3:

Die Anforderungen der Aufgabe führen zu folgender Belegung des Steuerregisters:

Bit	7	6	5	4	3	2	1	0
	0	1	0	0	0	0	1	1
	OE	IE	Modus, Interr.-Steuerung			ZM	T	

Also muß ins Steuerregister CR#3 der Hexadezimalwert: 0100 0011 = \$43 eingetragen werden.

Taste -:      \$F018   \$43

- II. **Laden der Auffangregister** von Zähler #3:

Taste A:      \$F01E   (\$xx)

Taste D:              \$FF

Taste +:      \$F01F   \$FF

- III. **Interrupt-Vektor zu Zähler #3 laden:** \$E4C0 nach \$002E, \$002F

Taste A:      \$002E   (\$E6)

Taste D:              \$E4

Taste +:              \$C0

- IV. **Starten des Zählers #3:**

Taste A:      \$F019   (\$00)

Taste D:              \$01      Vorselektion von Steuerregister CR#1

Taste -:      \$F018   \$00      Rücksetzen des „Startbits“ 0

Nach einer kurzen Zeit, die einem Zählzyklus entspricht, wird eine Unterbrechungsanforderung an die CPU gestellt. Es erscheint in der Anzeige z.B. die Information:

bP E205 A6

Das Monitorprogramm wurde mit einem Sprung in die Break-Routine „verlassen“. Dem Gesamtschaltplan des Systems im Anhang von Kapitel 3 können Sie entnehmen, daß der IRQ-Ausgang des Zeitgeber-/Zähler-Bausteins MC6840 mit dem NMI-Eingang (*Non Maskable Interrupt*) der CPU verbunden ist. Die Startadresse der NMI-Routine im Monitor ist in III. mit der Startadresse der Break-Routine vorbelegt worden. Ein Zählzyklus dauert:

$$2^{16} * 8 \text{ Taktzyklen von } E = 524,288 \text{ ms.}$$

## V. Assemblerprogramm

Die minimale Zählfrequenz wird für den maximalen Initialwert des Zählers, also für \$FFFF erreicht. Bei der gewählten Betriebsart erscheint am Ausgang O3 ein Rechtecksignal mit gleich langer Impulsdauer und Impulspause (*Square Wave*).

```

1      0000      HZ          EQU    $0000      ; Hexadezimalzähler
2      002E      INT_Z3     EQU    $002E      ; Interruptvektor Zähler #3
3      F018      SR_Z1      EQU    $F018      ; Steuerregister Zähler #1
4      F018      SR_Z3      EQU    $F018      ; Steuerregister Zähler #3
5      F019      SR_Z2      EQU    $F019      ; Steuerregister Zähler #2
6      F019      STREG       EQU    $F019      ; Statusregister
7      F01E      LATCH_Z3   EQU    $F01E      ; Auffangregister Zähler #3
8      F120      SHOWB7SG   EQU    $F120      ; Anzeigeroutine
9
10     0400                      ORG    $0400
11     0400  CC 06 00  START: LDD    #$0600
12     0403  DD 2E                      STD    INT_Z3      ; Interruptvektor laden
13     0405  7F F0 19                  CLR    SR_Z2      ; Steuerregister #3 selektieren
14     0408  86 C3                      LDA    #$C3      ; Steuerregister #3 mit $C3 =
15     040A  B7 F0 18                  STA    SR_Z3      ; 1100 0011 laden
16     040D  CC FF FF                  LDD    $FFFF      ; Auffangregister von Zähler #3
17     0410  FD F0 1E                  STD    LATCH_Z3    ; mit $FFFF laden
18     0413  86 01                      LDA    #$01      ; Steuerregister von Zähler #1
19     0415  B7 F0 19                  STA    SR_Z2      ; anwählen
20     0418  7F F0 18                  CLR    SR_Z1      ; Zähler starten
21     041B  7E 04 1B  L:             JMP    L              ; Endlosschleife
22
23     0600                      ORG    $0600      ; Interruptroutine
24     0600  0C 00      INT:         INC    HZ          ; Inkrementieren des
25     0602  8E 00 00                      LDX    #$0000      ; Anzeigestellen S1,S0
26     0605  D6 00                      LDB    HZ          selektieren
27     0607  BD F1 20                  JSR    SHOWB7SG; Zählerstand anzeigen
28     060A  F6 F0 19                  LDB    STREG      ; Löschen des Interrupt-Flags I
29     060D  FC F0 1E                  LDD    LATCH_Z3   ; durch Lesen der Register
30     0610  3B                      RTI              ; Rücksprung

```

**Zu P5.2-2:****I. „Programmieren“ des Zählers #3:**

Die Anforderungen der Aufgabe führen zu folgender Belegung des Steuerregisters:

Bit	7	6	5	4	3	2	1	0
	1	0	0	0	0	0	1	1
	OE	IE	Modus, Interr.-Steuerung			ZM	T	

Also muß ins Steuerregister CR#3 der Hexadezimalwert: 1000 0011 = \$83 eingetragen werden.

Taste A:	\$F019	(\$00)	Selektion des Steuerregisters
Taste D:		\$00	von Zähler #3
Taste -:	\$F018	\$83	Betriebsart wählen

**II. Berechnung von Frequenz und Periodendauer:**

Zykluslänge des Taktes E: 1 µs

Periodendauer = 2 \* (Inhalt des Auffangregisters +1) \* 8 \* 1 µs

$$\approx \left\{ \begin{array}{l} 2 * 8 * 4.096 \quad \text{ms} = 65.54 \quad \text{ms} \\ 2 * 8 * 32.768 \quad \text{ms} = 524.29 \quad \text{ms} \\ 2 * 8 * 4.096 * 15 \quad \text{ms} = 983.04 \quad \text{ms} \end{array} \right.$$

(Auf die maßstäbliche Darstellung der Rechtecksignale wird hier verzichtet.)

**III. Programmierung des Bausteins: (Fortsetzung von I.)**

Taste A:	\$F019	(\$00)	Selektion des Steuerregisters
Taste D:		\$00	von Zähler #3
Taste -:	\$F018	\$83	Betriebsart wählen
Taste A:	\$F01E		
Taste D:		\$XX	\$XXXX Initialwert
Taste +:	\$F01F	\$XX	
Taste A:	\$F019	(\$00)	Selektion des Steuerregisters
Taste D:		\$01	von Zähler #1
Taste -:	\$F018	\$00	Starten der Zähler

**IV. Änderung der Parameter:**

- Setzen des Gate-Eingangs G3 auf H-Potential stoppt den Zähler, Rücksetzen auf L läßt ihn weiterlaufen.
- Nach dem Einschreiben eines neuen Wertes in das Auffangregister wird der Zähler neu initialisiert und mit dem neuen Wert betrieben. Achtung: Erst das Einschreiben des Wertes in \$F01F bewirkt diese Änderung !

- Eine Änderung des Steuerwortes veranlaßt unmittelbar die Ausführung in der neuen Betriebsart. Z.B. erhöht eine Änderung des Wertes \$83 zu \$82, also eine Inversion des Bits 0, die Ausgangsfrequenz um den Faktor 8, da nun der Frequenzteiler 1:8 deaktiviert ist. Wichtig: Zunächst wieder das Steuerregister von Zähler #3 selektieren:

Taste A:      \$F019    (\$00)    Selektion des Steuerregisters  
 Taste D:                \$00            von Zähler #3  
 Taste -:      \$F018    \$82        neues Steuerwort eingeben

### Zu P5.2-3:

#### I. „Programmieren“ des Zählers #3:

Die Anforderungen der Aufgabe führen zu folgender Belegung des Steuerregisters:

Bit	7	6	5	4	3	2	1	0
	1	0	0	0	0	0	0	0
	OE	IE	Modus, Interr.-Steuerung			ZM	T	

Also muß ins Steuerregister CR#3 der Hexadezimalwert: 1000 0000 = \$80 eingetragen werden.

Taste A:      \$F019    (\$00)    Selektion des Steuerregisters  
 Taste D:                \$00            von Zähler #3  
 Taste -:      \$F018    \$80        Betriebsart wählen

#### II. Berechnung der Triggerfrequenz und des Ausgangssignals:

100 Hz entsprechen einer Schwingungsdauer von 10 ms. also einer Impulsdauer und Pausendauer von je 5 ms. Daher müssen die Initialwerte in den Registern U,Y für das Programm aus P4.2-3, das sich der Routine DLY1MS bedient, auf den Wert \$5 gesetzt werden.

$$\text{Periodendauer des Ausgangssignals} \approx 2 * (\$100+1) * \text{Schwingungsdauer} \\ = 5.14 \text{ s}$$

Programmierung des Bausteins: (Fortsetzung von I.)

Taste A:      \$F01E    \$01        Laden des Auffangregisters  
 Taste +:      \$F01F    \$00            von Zähler #3  
 Taste A:      \$F019    (\$00)  
 Taste D:                \$01        Steuerregister CR#1 selektieren  
 Taste -:      \$F018    \$00        Starten der Zähler

#### III. Berechnung der 1-Hz-Frequenz:

1 Hz bei einer Grundschiebungsdauer von 10 ms bedeutet, daß ein Zählzyklus 100 Takte, also Impulsdauer und Impulspause jeweils 50 Takte, dauern müssen. Daraus ergibt sich ein Initialwert von 50 = \$32.

Taste A:      \$F01E    \$00      Laden des Auffangregisters  
 Taste +:      \$F01F    \$32      von Zähler #3

#### IV. Berechnung der maximalen Frequenz:

Initialwert \$0000 ergibt die minimale Schwingungsdauer:

$2 * (0+1) * 10 \text{ ms} = 20 \text{ ms}$ . Daraus ergibt sich eine maximale Frequenz von 50 Hz.

#### V. Auswirkung von G3:

Für  $\overline{G3} = H$  wird der Zähler gestoppt, die Schaltung funktioniert also dann nicht mehr.

#### Zu P5.2-4:

##### I. „Programmieren“ des Zählers #3:

Die Anforderungen der Aufgabe führen zu folgender Belegung des Steuerregisters:

Bit	7	6	5	4	3	2	1	0
	1	0	1	0	0	0	1	1
	OE	IE	Modus, Interr.-Steuerung			ZM	T	

Also muß ins Steuerregister CR#3 der Hexadezimalwert: 1010 0011 = \$A3 eingetragen werden.

##### II. Programm: Starten durch die Software

(für den 6809-Simulator oder Praktikumsrechner)

```

1   F018      SR_Z1      EQU    $F018      ; Steuerregister Zähler #1
2   F018      SR_Z3      EQU    $F018      ; Steuerregister Zähler #3
3   F019      SR_Z2      EQU    $F019      ; Steuerregister Zähler #2
4   F01E      LATCH_Z3   EQU    $F01E      ; Auffangregister Zähler #3
5   F003      CRB        EQU    $F003      ; Steuerregister Port B
6   F110      CLRDISP    EQU    $F110      ; Anzeige löschen
7   F140      KEY        EQU    $F140      ; Tastatur abfragen
8   F143      HALTKEY    EQU    $F143      ; Tastatur abfragen mit Halt
9   F156      SHOWADR    EQU    $F156      ; Adresse einlesen und anz.
10  F160      DLY1MS     EQU    $F160      ; Verzögerung
11
12 0400                                ORG    $0400
13 0400 7F F0 19                    CLR    SR_Z2 ; Steuerreg. Zähler #3 selektieren
14 0403 86 A3                        LDA    #$A3 ; gewünschte Betriebsart
15 0405 B7 F0 18                    STA    SR_Z3 ; programmieren
16 0408 CC FF FF                    LDD    #$FFFF ; maximale Impulsdauer
17 040B FD F0 1E                    STD    LATCH_Z3 ; laden

```

```

18 040E 86 01      LDA  #$01      ; Steuerregister Zähler #1
19 0410 B7 F0 19      STA  SR_Z2      ; selektieren
20 0413 7F F0 18      CLR  SR_Z1      ; (alle) Zähler starten
21 0416 10 8E 00 40    LDY  #$0040    ; Y initialisieren
22 041A BD F1 10      JSR  CLRDISP    ; Anzeige löschen
23
24 041D BD F1 40    LOOP JSR  KEY      ; Tastatur abfragen
25 0420 C1 86      CMPB #$86      ; 'S' gedrückt ?
26 0422 27 18      BEQ  L1        ; falls ja, neuen Wert einlesen
27 0424 CC FF FF      LDD  #$FFFF    ; max. Zählzykluslänge vorgeben
28 0427 FD F0 1E      STD  LATCH_Z3 ; Initialisieren des Zählers #3
29 042A 86 34      LDA  #$34      ; LED an CB2
30 042C B7 F0 03      STA  CRB      ; ausschalten
31 042F BD F1 60      JSR  DLY1MS    ; Verzögerung um ca. (Y) ms
32 0432 86 3C      LDA  #$3C      ; LED an CB2
33 0434 B7 F0 03      STA  CRB      ; anschalten
34 0437 BD F1 60      JSR  DLY1MS    ; Verzögerung um ca. (Y) ms
35 043A 20 E1      BRA  LOOP
36
37 043C BD F1 56    L1   JSR  SHOWADR ; Einlesen des neuen Wertes
38 043F 20 DC      BRA  LOOP

```

### III. Programm: Starten durch die Hardware: (nur für den Praktikumsrechner)

Zunächst müssen Sie den Ausgang CB2 mit dem Eingang G3 des Zählers verbinden.

Bit	7	6	5	4	3	2	1	0
	1	0	1	1	0	0	1	1
	OE	IE	Modus, Interr.-Steuerung		ZM	T		

Also muß ins Steuerregister CR#3 nun der Hexadezimalwert: 1011 0011 = \$B3 eingetragen werden.

Änderungen im Programm aus II.:

- 14 0403 86 B3                      LDA #\$B3            ; Betriebsart wählen
- Die Zeilen 27 und 28 müssen gelöscht werden.

**Zu P5.2-5:****„Programmieren“ des Zählers #3:**

Die Anforderungen der Aufgabe führen zu folgender Belegung des Steuerregisters:

Bit	7	6	5	4	3	2	1	0
	1	1	0	1/0	1	0	1	0
	OE	IE	Modus, Interr.-Steuerung			ZM	T	

Also muß ins Steuerregister CR#3 der Hexadezimalwert:

1100 1010 = \$CA für die Schwingungsdauer-Messung

1101 1010 = \$DA für die Impulsdauer-Messung

eingetragen werden.

**I./II. Programm:**

```

1      002E      INT_Z3      EQU    $002E      ; Interruptvektor Z#3
2      F003      CRB         EQU    $F003      ; Steuerregister Port B
3      F018      SR_Z1      EQU    $F018      ; Steuerregister Zähler #1
      F018      SR_Z3      EQU    $F018      ; Steuerregister Zähler #3
4      F019      SR_Z2      EQU    $F019      ; Steuerregister Zähler #2
5      F019      STREG      EQU    $F019      ; Statusregister aller Zähler
6      F01E      LATCH_Z3   EQU    $F01E      ; Auffangregister Zähler #3
7      F110      CLRDISP    EQU    $F110      ; Anzeige löschen
8      F123      SHOWD7SG   EQU    $F123      ; D in Anzeige bringen
9      F140      KEY        EQU    $F140      ; Tastatur abfragen
10     F156      SHOWADR     EQU    $F156      ; Adresse eingeben/anzeig.
11     F160      DLY1MS     EQU    $F160      ; Verzögerung
12
13     0400                      ORG    $0400
14     0400  CC 05 00      START  LDD    #$0500 ;
15     0403  DD 2E                      STD    INT_Z3 ; Interruptvektor setzen
16     0405  7F F0 19      CLR     SR_Z2 ; Steuerregister Zähler #3 selekt.
17     0408  86 CA                      LDA    #$CA ; Schwingungsdauer,
                                           ; #$DA: Impulsdauer !!
18     040A  B7 F0 18      STA     SR_Z3 ; Betriebsart wählen
19     040D  CC FF FF      LDD     $FFFF ; max. Zykluslänge
20     0410  FD F0 1E      STD     LATCH_Z3 ; programmieren
21     0413  86 01                      LDA    #$01 ; Steuerregister Zähler #1
22     0415  B7 F0 19      STA     SR_Z2 ; selektieren
23     0418  7F F0 18      CLR     SR_Z1 ; alle Zähler starten
24     041B  BD F1 10      JSR     CLRDISP ; Anzeige löschen
25     041E  86 01      EING     LDA    #$01 ; (alle) Zähler

```

```

26 0420 B7 F0 18      STA  SR_Z3    ;  stoppen
27 0423 8E 00 04      LDX  #$0004    ;  neuen Wert eingeben und
28 0426 BD F1 56      JSR  SHOWADR;    anzeigen
29 0429 1F 20          TFR  Y,D      ;  (überflüssige) Ausgabe
30 042B BD F1 23      JSR  SHOWD7SG ;  des neuen Wertes
31 042E 7F F0 18      CLR  SR_Z3    ;  (alle) Zähler starten
32 0431 BD F1 40      RE_ECK JSR  KEY      ;  Tastatur ohne Halt abfragen
33 0434 C1 86          CMPB #$86     ;  Taste 'S' für neue Eingabe
34 0436 27 E6          BEQ  EING     ;  neuer Wert gewünscht
35 0438 86 34          LDA  #$34     ;  LED an CB2
36 043A B7 F0 03      STA  CRB      ;  ausschalten
37 043D BD F1 60      JSR  DLY1MS   ;  Verzögerung
38 0440 86 3C          LDA  #$3C     ;  LED an CB2
39 0442 B7 F0 03      STA  CRB      ;  einschalten
40 0445 BD F1 60      JSR  DLY1MS   ;  Verzögerung
41 0448 20 E7          BRA  RE_ECK
42
43 0500                ORG  $0500
44 0500 8E 00 00      INT  LDX  #$0      ;  Ausgabe des Meßwerts
45 0503 B6 F0 19      LDA  STREG    ;  Löschen des I-Flags
46 0506 CC FF FF      LDD  $FFFF    ;  Berechnung des Meßwertes
47 0509 B3 F0 1E      SUBD LATCH_Z3;  aus aktuellem Zähler
48 050C BD F1 23      JSR  SHOWD7SG; Anzeige des Meßwertes
49 050F 3B            RTI

```



Bei den Lösungen der folgenden Aufgaben haben wir - bis auf wenige wichtige Zeilen - bewußt auf die Angabe von Kommentaren verzichtet. Sie sollen sich selbst darüber klar werden, was in den einzelnen Zeilen geschieht und dazu ggf. die Trace- und Register-Funktionen des Monitors benutzen.

### Zu P5.2-6:

1	F002	DDRB	EQU	\$F002
2	F002	DRB	EQU	\$F002
3	F003	CRB	EQU	\$F003
4	F110	CLRDISP	EQU	\$F110
5	F123	SHOWD7SG	EQU	\$F123
6	F120	SHOWB7SG	EQU	\$F120
7	F160	DLY1MS	EQU	\$F160
8	002E	INT_Z3	EQU	\$002E
9	0038	INT_CB1	EQU	\$0038
10	F018	SR_Z3	EQU	\$F018
11	F019	SR_Z2	EQU	\$F019
12	F019	STREG	EQU	\$F019
13	F01E	LATCH_Z3	EQU	\$F01E
14	0000	HZ	EQU	\$0

15	<b>Hauptprogramm</b>			
16	0400	ORG	\$0400	
17	0400	CC 04 80	START	LDD #\$0480
18	0403	DD 2E		STD INT_Z3
19	0405	CC 04 C0		LDD #\$04C0
20	0408	DD 38		STD INT_CB1
21	040A	0F 00		CLR HZ
22	040C	0F 01		CLR HZ+1
23	040E	0F 02		CLR HZ+2
24	0410	7F F0 19		CLR SR_Z2
25	0413	86 C2		LDA #\$C2
26	0415	B7 F0 18		STA SR_Z3
27	0418	CC FF FF		LDD \$FFFF
28	041B	FD F0 1E		STD LATCH_Z3
29	041E	86 01		LDA \$01
30	0420	B7 F0 19		STA SR_Z2
31	0423	BD F1 10		JSR CLRDISP
32	0426	C6 34		LDB \$34
33	0428	F7 F0 03		STB CRB
34	042B	7E 04 40		JMP RECHT
35	<b>Endlosschleife</b>			
36	0440	ORG	\$0440	
38	0440	10 8E 02 00	RECHT	LDY \$0200
39	0444	C6 37	LOOP	LDB \$37

```

40 0446 F7 F0 03      STB  CRB
41 0449 BD F1 60      JSR  DLY1MS
42 044C C6 3F         LDB  #$3F
43 044E F7 F0 03      STB  CRB
44 0451 BD F1 60      JSR  DLY1MS
45 0454 7E 04 44      JMP  LOOP
46
47 0480                ORG  $0480
48 0480 0C 00      INT-Z3  INC  HZ
49 0482 F6 F0 19      LDB  STREG
50 0485 FC F0 1E      LDD  LATCH_Z3
51 0488 3B           RTI
52
53 04C0                ORG  $04C0
54 04C0 0D 02      INT-CB2  TST  HZ+2
55 04C2 26 09      BNE  L1
56 04C4 0C 02      INC  HZ+2
57 04C6 7F F0 18      CLR  SR_Z3 ; Timer Starten
58 04C9 F6 F0 02      LDB  DRB
59 04CC 3B           RTI
60 04CD F6 F0 19      L1    LDB  STREG ; INT loeschen
61 04D0 FE F0 1E      LDU  LATCH_Z3
62 04D3 C6 01      LDB  #$1
63 04D5 F7 F0 18      STB  SR_Z3 ; Timer Stoppen
64 04D8 8E 00 04      LDX  #4
65 04DB D6 00      LDB  HZ
66 04DD BD F1 20      JSR  SHOWB7SG
67 04E0 8E 00 06      LDX  #$6
68 04E3 0C 01      INC  HZ+1
69 04E5 D6 01      LDB  HZ+1
70 04E7 BD F1 20      JSR  SHOWB7SG
71 04EA 8E 00 00      LDX  #0
72 04ED DF 03      STU  HZ+3
73 04EF CC FF FF      LDD  #$FFFF
74 04F2 93 03      SUBD HZ+3
75 04F4 BD F1 23      JSR  SHOWD7SG
76 04F7 0F 00      CLR  HZ
77 04F9 0F 02      CLR  HZ+2
78 04FB F6 F0 02      LDB  DRB
79 04FE 3B           RTI

```

**Zu P5.3-1:**

1	F110	CLRDISP	EQU	\$F110	
2	F140		KEY	EQU	\$F140
3	F143		HALTKEY	EQU	\$F143
4	F123		SHOWD7SG	EQU	\$F123
5	F156		SHOWADR	EQU	\$F156
6	F160		DLY1MS	EQU	\$F160
7					
8	0400		ORG	\$0400	
9	0400 BD F1 10	START	JSR	CLRDISP	
10	0403 8E 00 02		LDX	#\$0002	
11	0406 4F		CLRA		
12	0407 5F		CLRB		
13	0408 34 06		PSHS	D	
14	040A C6 01		LDB	#1	
15	040C D7 00		STB	\$0	
16	040E 0F 01		CLR	\$1	
17					
18	0410 35 06	LOOP	PULS	D	
19	0412 BD F1 23		JSR	SHOWD7SG	
20	0415 10 8E 00 10		LDY	#\$0010	
21	0419 BD F1 60		JSR	DLY1MS	
22	041C 1E 89		EXG	A,B	
23	041E 9B 00		ADDA	\$0	
24	0420 0D 01		TST	\$1	
25	0422 27 01		BEQ	HEX	
26	0424 19		DAA		
27	0425 1E 89	HEX	EXG	A,B	
28	0427 89 00		ADCA	#0	
29	0429 0D 01		TST	\$1	
30	042B 27 01		BEQ	WEITER	
31	042D 19		DAA		
32	042E 34 06	WEITER	PSHS	D	
33					
34	0430 BD F1 40	ABFRAGE	JSR	KEY	
35	0433 C1 86		CMPB	#\$86	
36	0435 27 F9		BEQ	ABFRAGE	
37	0437 C1 84		CMPB	#\$84	
38	0439 27 C5		BEQ	START	
39	043B C1 82		CMPB	#\$82	
40	043D 26 09		BNE	L3	
41	043F 35 06		PULS	D	
42	0441 BD F1 56		JSR	SHOWADR	
43	0444 34 20		PSHS	Y	
44	0446 20 C8		BRA	LOOP	

45	0448	C1 80	L3	CMPB	#\$80
46	044A	26 04		BNE	L1
47	044C	0F 01		CLR	\$1
48	044E	20 C0		BRA	LOOP
49	0450	C1 81	L1	CMPB	#\$81
50	0452	26 06		BNE	L2
51	0454	C6 01		LDB	#1
52	0456	D7 01		STB	\$1
53	0458	20 B6		BRA	LOOP
54	045A	C1 00	L2	CMPB	#\$00
55	045C	2B B2		BMI	LOOP
56	045E	D7 00		STB	\$0
57	0460	20 AE		BRA	LOOP
58	0462		END		

**Zu P5.3-2:**

1	0400	ORG	\$0400
2	0400	START	JSR CLRDISP
3	0403	ANFANG	LDX #5
4	0406		LDA #2
5	0408	TASTE	JSR HALTKEY
6	040B		CMPB #A
7	040D		BHS TASTE
8	040F		STB , -X
9	0411		JSR SHOWT7SG
10	0414		DECA
11	0415		BPL TASTE
13	0417		LDA \$4
14	041A		CMPA #1
15	041C		BHI FALSCH
16	041E		LDB #64
17	0420		MUL
18	0422		STB \$0
19	0423		LDA \$3
20	0425		LDB #A
21	0427		MUL
22	0429		ADDB \$0
23	042A		ADDB \$2
24	042C		STB \$0
25	042E		CMPB #80
26	0430		BHS FALSCH
27	0432	TASTE1	JSR HALTKEY
29	0434		CMPB #86
30	0437		BEQ START

31	0439	C1 80		CMPB	#\$80
32	043B	26 11		BNE	MINUS
33	043D	D6 00		LDB	\$0
34	043F	86 70		LDA	#\$70
35	0441	8E 00 00	DARST	LDX	#0
36	0443	BD F1 20		JSR	SHOWB7SG
37	0446	30 07		LEAX	7,X
38	0448	BD F1 13		JSR	SHOWA
39	044B	20 E4		BRA	TASTE1
41	044D	C1 81	MINUS	CMPB	#\$81
42	044F	26 E0		BNE	TASTE1
43	0451	D6 00		LDB	\$0
44	0453	C8 FF		EORB	#\$FF
45	0455	5C		INCB	
46	0456	86 40		LDA	#\$40
47	0458	20 EA		BRA	DARST
49	045A	10 8E 03 E8	FALSCH	LDY	#\$03E8
50	045E	BD F1 60		JSR	DLY1MS
51	0461	20 9D		BRA	START
52	0463		END		
53	F160		DLY1MS	EQU	\$F160
54	F110		CLRDISP	EQU	\$F110
55	F120		SHOWB7SG	EQU	\$F120
56	F143		HALTKEY	EQU	\$F143
57	F116		SHOWD	EQU	\$F116
58	F113		SHOWA	EQU	\$F113
59	F11C		SHOWT7SG	EQU	\$F11C

**Zu P5.3-3:**

1	F100		T7SG	EQU	\$F100
2	F110		CLRDISP	EQU	\$F110
3	F133		SHOWDBUF	EQU	\$F133
4	F139		RLDBUF	EQU	\$F139
5	F13C	COPYDBUF		EQU	\$F13C
6	F140		KEY	EQU	\$F140
7	F143		HALTKEY	EQU	\$F143
8	F113		SHOWA	EQU	\$F113
9	F11C	SHOWT7SG		EQU	\$F11C
10	F156		SHOWADR	EQU	\$F156
11	F160		DLY1MS	EQU	\$F160
12					
13	0600		ORG	\$0600	
14	0600	40	TABELLE	DB	\$40

15	0601	79		DB	\$79
16	0602	24		DB	\$24
17	0603	30		DB	\$30
18	0604	19		DB	\$19
19	0605	12		DB	\$12
20	0606	02		DB	\$02
21	0607	78		DB	\$78
22	0608	00		DB	\$00
23	0609	10		DB	\$18
24	060A	08		DB	\$08
25	060B	03		DB	\$03
26	060C	27		DB	\$27
27	060D	21		DB	\$21
28	060E	06		DB	\$06
29	060F	0E		DB	\$0E
30	0610				
31					
32	0400		ORG	\$0400	
33	0400	BD F1 10	EINGABE	JSR	CLRDISP
34	0403	CE 06 00		LDU	#\$0600
35	0406	8E 00 05		LDX	#\$5
36	0409	BD F1 43	L2	JSR	HALTKEY
37	040C	2B FB		BMI	L2
38	040E	E7 84		STB	,X
39	0410	BD F1 1C		JSR	SHOWT7SG
40	0413	30 1F		LEAX	-1,X
41	0415	8C 00 01		CMPX	#1
42	0418	26 EF		BNE	L2
43					
44	041A	BD F1 43	WEITER	JSR	HALTKEY
45	041D	C1 86		CMPB	#\$86
46	041F	27 DF		BEQ	EINGABE
47	0421	C1 82		CMPB	#\$84
48	0423	26 04		BNE	L1
49	0425	D7 01		STB	\$1
50	0427	20 06		BRA	L4
51	0429	C4 03	L1	ANDB	#\$3
52	042B	D7 00		STB	\$0
53	042D	D7 01		STB	\$1
54					
55	042F	8E 00 05	L4	LDX	#\$5
56	0432	E6 84	INVERT	LDB	,X
57	0434	0D 00		TST	\$0
58	0436	27 07		BEQ	PLUS
59					
60	0438	A6 C5	MINUS	LDA	B,U

61	043A	BD F1 13		JSR	SHOWA
62	043D	20 03		BRA	NEXT
63	043F	BD F1 1C	PLUS	JSR	SHOWT7SG
64					
65	0442	30 1F	NEXT	LEAX	-1,X
66	0444	8C 00 01		CMPX	#1
67	0447	26 E9		BNE	INVERT
68					
69	0449	0D 01		TST	\$1
70	044B	2A CD		BPL	WEITER
71	044D	10 8E 01 00		LDY	#\$0100
72	0451	BD F1 60		JSR	DLY1MS
73	0454	96 00		LDA	\$0
74	0456	88 FF		EORA	#\$FF
75	0458	84 01		ANDA	#\$1
76	045A	97 00		STA	\$0
77	045C	BD F1 40		JSR	KEY
78	045F	C1 83		CMPB	#\$82
79	0461	26 CC		BNE	L4
80	0463	20 B5		BRA	WEITER
81	0465			END	

**Zu P5.3-4:**

1	F110		CLRDISP	EQU	\$F110
2	F143		HALTKEY	EQU	\$F143
3	F140		KEY	EQU	\$F140
4	F120		SHOWB7SG	EQU	\$F120
5	F123		SHOWD7SG	EQU	\$F123
6	F156		SHOWADR	EQU	\$F156
7	F113		SHOWA	EQU	\$F113
8	F116	S	HOWD	EQU	\$F116
9	F11C	SHOWT7SG		EQU	\$F11C
10	F160		DLY1MS	EQU	\$F160
11	0000	HZ0	EQU	\$0	
12	0001	HZ1	EQU	\$1	
13					
14	0400		ORG	\$0400	
15	0400	BD F1 10	START	JSR	CLRDISP
16	0403	8E 00 02		LDX	#2
17	0406	BD F1 56		JSR	SHOWADR
18	0409	C1 82		CMPB	#\$82
19	040B	26 F3		BNE	START
20	040D	1F 23		TFR	Y,U
21	040F	C6 01		LDB	#1

22	0411	D7 00		STB	HZ0
23					
24	0413	D6 00	L4	LDB	HZ0
25	0415	8E 00 00		LDX	#0
26	0418	BD F1 20		JSR	SHOWB7SG
27	041B	8E 00 02		LDX	#2
28	041E	BD F1 56		JSR	SHOWADR
29					
30	0421	C1 80	L1	CMPB	#\$80
31	0423	26 06		BNE	L2
32					
33			PLUS	; Taste '+'	
34	0425	36 20		PSHU	Y
35	0427	0C 00		INC	HZ0
36	0429	20 E8		BRA	L4
37					
38	042B	C1 81	L2	CMPB	#\$81
39	042D	27 05		BEQ	MINUS
40	042F	BD F1 43		JSR	HALTKEY
41	0432	20 DF		BRA	L4
42					
43			MINUS	; Taste '-'	
44	0434	0A 00		DEC	HZ0
45	0436	27 19		BEQ	L10
46	0438	D6 00		LDB	HZ0
47	043A	8E 00 00		LDX	#0
48	043D	BD F1 20		JSR	SHOWB7SG
49	0440	8E 00 02		LDX	#2
50	0443	37 06		PULU	B,A
51	0445	BD F1 23		JSR	SHOWD7SG
52	0448	BD F1 43		JSR	HALTKEY
53	044B	C1 81		CMPB	#\$81
54	044D	27 E5		BEQ	MINUS
55	044F	20 C2		BRA	L4
56					
57	0451	C6 50	L10	LDB	#\$50
58	0453	8E 00 06		LDX	#6
59	0456	BD F1 10		JSR	CLRDISP
60	0459	BD F1 20		JSR	SHOWB7SG
61	045C	10 8E 01 00	L7	LDY	#\$03E8
62	0460	BD F1 60		JSR	DLY1MS
63	0463	20 9B		BRA	START
64	0465		END		



**Zu P5.3-5:**

1	F110	CLRDISP	EQU	\$F110
2				
3	F140	KEY	EQU	\$f140
4	F143	HALTKEY	EQU	\$F143
5	F11C	SHOWT7SG	EQU	\$F11C
6	F120	SHOWB7SG	EQU	\$F120
7	F123	SHOWD7SG	EQU	\$F123
8	F156	SHOWADR	EQU	\$F156
9	F113	SHOWA	EQU	\$F113
10	F116	SHOWD	EQU	\$F116
11	F160	DLY1MS	EQU	\$F160
12	0000	HZ	EQU	\$10
13				
14	0400		ORG	\$0400
15	0400 BD F1 10	START	JSR	CLRDISP
16	0403 10 8E 01 00		LDY	#\$01F4
17				
18	0407 8E 00 01	L1	LDX	#1
19	040A 1F 89		TFR	A,B ; Default A=\$00
20	040C 7F F0 20		CLR	\$F020
21	040F 7F F0 28		CLR	\$F028
22	0412 BD F1 1C		JSR	SHOWT7SG
23	0415 BD F1 43		JSR	HALTKEY
24	0418 30 1F		LEAX	-1,X
25	041A C4 0F		ANDB	#\$0F
26	041C E7 C6		STB	A,U ; Default U=\$0000
27	041E BD F1 1C		JSR	SHOWT7SG
28	0421 BD F1 60		JSR	DLY1MS
29	0424 4C	INCA		
30	0425 81 10		CMPA	#\$10
31	0427 26 DE		BNE	L1
32				
33	0429 CE 04 00		LDU	#\$0400
34	042C 8E 00 02	L2	LDX	#2
35	042F 1F 30		TFR	U,D
36	0431 BD F1 23		JSR	SHOWD7SG
37	0434 30 1E		LEAX	-2,X
38	0436 E6 C0		LDB	,U+
39	0438 BD F1 20		JSR	SHOWB7SG
40	043B 1F 98		TFR	B,A
41	043D C4 0F		ANDB	#\$F
42	043F E6 85		LDB	B,X
43	0441 D7 00		STB	\$0

44	0443	44		LSRA	
45	0444	44		LSRA	
46	0445	44		LSRA	
47	0446	44		LSRA	
48	0447	E6 86		LDB	A,X
49	0449	58		ASLB	
50	044A	58	ASLB		
51	044B	58	ASLB		
52	044C	58	ASLB		
53	044D	DA 00		ORB	\$0
54	044F	30 06		LEAX	6,X
55	0451	BD F1 20		JSR	SHOWB7SG
56	0454	BD F1 60		JSR	DLY1MS
57	0457	BD F1 40		JSR	KEY
58	045A	C1 86		CMPB	#\$86
59	045C	26 CE		BNE	L2
60	045E	3F	SWI		
61					
62	045F			END	

**Zu P5.3-6:**

1	F110	CLRDISP	EQU	\$F110
2				
3	F140	KEY	EQU	\$f140
4	F143	HALTKEY	EQU	\$F143
5	F11C	SHOWT7SG	EQU	\$F11C
6	F120	SHOWB7SG	EQU	\$F120
7	F123	SHOWD7SG	EQU	\$F123
8	F156	SHOWADR	EQU	\$F156
9	F150	SHOWDATA	EQU	\$F150
10	F113	SHOWA	EQU	\$F113
11	F116	SHOWD	EQU	\$F116
12	F160	DLY1MS	EQU	\$F160
13	0000	HZ	EQU	\$0
14				
15	0400		ORG	\$0400
16	0400 BD F1 10	START	JSR	CLRDISP
17	0403 10 8E 01 00		LDY	#\$0100
18	0407 CC 01 23		LDD	#\$0123
19	040A CE 00 00		LDU	#0
20	040D 8E 00 04	L1	LDX	#4
21	0410 BD F1 23		JSR	SHOWD7SG
22	0413 C3 44 44		ADDD	#\$4444
23	0416 34 06		PSHS	B,A

24	0418	30 1C		LEAX	-4,X
25	041A	BD F1 56		JSR	SHOWADR
26	041D	10 AF C1		STY	,U++
27	0420	35 06		PULS	B,A
28	0422	11 83 00 08		CMPU	#8
29	0426	26 E5		BNE	L1
30					
31	0428	10 8E 01 00		LDY	#\$0100
32	042C	BD F1 10	L2	JSR	CLRDISP
33	042F	8E 00 00		LDX	#0
34	0432	BD F1 50		JSR	SHOWDATA
35	0435	97 11		STA	\$11
36	0437	84 0F		ANDA	#\$F
37	0439	44		LSRA	
38	043A	E6 86		LDB	A,X
39	043C	25 04		BCS	L20
40	043E	54	LSRB		
41	043F	54		LSRB	
42	0440	54		LSRB	
43	0441	54		LSRB	
44	0442	C4 0F	L20	ANDB	#\$F
45	0444	D7 10		STB	\$10
46					
47	0446	96 11		LDA	\$11
48	0448	44		LSRA	
49	0449	44		LSRA	
50	044A	44	LSRA		
51	044B	44	LSRA		
52	044C	44	LSRA		
53	044D	E6 86		LDB	A,X
54	044F	24 04		BCC	L30
55	0451	58		LSLB	
56	0452	58		LSLB	
57	0453	58		LSLB	
58	0454	58		LSLB	
59	0455	C4 F0	L30	ANDB	#\$F0
60	0457	DA 10		ORB	\$10
61					
62	0459	8E 00 06		LDX	#6
63	045C	BD F1 20		JSR	SHOWB7SG
64	045F	BD F1 60		JSR	DLY1MS
65	0462	20 C8		BRA	L2
66					
67	0464			END	

**Zu P5.3-7:**

1	F110	CLRDISP	EQU	\$F110
2	F143	HALTKEY	EQU	\$F143
3	F120	SHOWB7SG	EQU	\$F120
4	F123	SHOWD7SG	EQU	\$F123
5	F156	SHOWADR	EQU	\$F156
6	F113	SHOWA	EQU	\$F113
7	F116	SHOWD	EQU	\$F116
8	F160	DLY1MS	EQU	\$F160
9	0000	HZ	EQU	\$0
10				
11	0400	ORG	\$0400	
12	0400 BD F1 10	START	JSR	CLRDISP
13	0403 CE 10 00		LDU	#\$1000
14	0406 10 8E 01 00		LDY	#\$0100
15	040A BD F1 43	L1	JSR	HALTKEY
16	040D 5D		TSTB	
17	040E 2B 07		BMI	L4
18	0410 36 04		PSHU	B
19	0412 BD 06 00		JSR	UP
20	0415 20 F3		BRA	L1
21				
22	0417 C1 80	L4	CMPB	#\$80
23	0419 27 1A		BEQ	PLUS
24	041B C1 81		CMPB	#\$81
25	041D 26 EB		BNE	L1
26				
27	041F 34 40	MINUS	PSHS	U ; '-'
28	0421 11 83 10 00	L6	CMPU	#\$1000
29	0425 27 0A		BEQ	L10
30	0427 37 04		PULU	B
31	0429 BD 06 00		JSR	UP
32	042C BD F1 60		JSR	DLY1MS
33	042F 20 F0		BRA	L6
34	0431 35 40	L10	PULS	U
35	0433 20 D5		BRA	L1
36				
37	0435 11 83 10 00	PLUS	CMPU	#\$1000 ; '+'
38	0439 27 CF		BEQ	L1
39	043B DF 00		STU	HZ
40	043D CE 10 00		LDU	#\$1000
41	0440 E6 C2	L5	LDB	,-U
42	0442 BD 06 00		JSR	UP
43	0445 BD F1 60		JSR	DLY1MS
44	0448 11 93 00		CMPU	HZ

```

45 044B 22 F3          BHI    L5
46 044D 20 BB          BRA    L1
47
48
49 0600                ORG     $600
50 0600 BD F1 10        UP     JSR     CLRDISP
51 0603 8E 00 00        LDX     #0
52 0606 86 40          LDA     #$40
53 0608 C4 07          ANDB    #$07
54 060A 3A             L2      ABX
55 060B BD F1 13        L3      JSR     SHOWA
56 060E 30 1F          LEAX    -1,X
57 0610 8C FF FF        CMPX    #$FFFF
58 0613 26 F6          BNE     L3
59 0615 39             RTS
60 0616
61
62 0616                END

```

**Zu P5.3-8:**

```

1    F110                CLRDISP    EQU    $F110
2    F11C                SHOWT7SG    EQU    $F11C
3    F133                SHOWDBUF    EQU    $F133
4    F130                CLDBUF      EQU    $F130
5    F140                KEY         EQU    $F140
6    F143                HALTKEY     EQU    $F143
7    F160                DLY1MS      EQU    $F160
8    0010                MODE        EQU    $0010
9
10   0400                ORG         $0400
11   0400 BD F1 10        START      JSR     CLRDISP
12   0403 C6 0E          LDB         #$E
13   0405 BD F1 1C          JSR     SHOWT7SG
14   0408 BD F1 43        MODUS      JSR     HALTKEY
15   040B C1 82          CMPB        #$82
16   040D 26 04          BNE         L3
17   040F 86 01          LDA         #$01
18   0411 20 06          BRA         L5
19   0413 C1 83        L3      CMPB        #$83
20   0415 26 F1          BNE         MODUS
21   0417 86 40          LDA         #$40
22   0419 97 10        L5      STA         MODE
23   041B BD F1 30          JSR     CLDBUF
24   041E 86 08          LDA         #$8
25   0420 BD F1 40        L1      JSR     KEY

```

26	0423	C1 FF		CMPB	#\$FF
27	0425	27 F9		BEQ	L1
28	0427	C1 80		CMPB	#\$80
29	0429	26 14		BNE	NPLUS
30					
31	042B	4D		TSTA	
32	042C	27 F2		BEQ	L1
33	042E	4A		DECA	
34	042F	D6 10		LDB	MODE
35	0431	E7 86		STB	A,X
36	0433	BD F1 33	L2	JSR	SHOWDBUF
37	0436	10 8E 00 10		LDY	#\$0010
38	043A	BD F1 60		JSR	DLY1MS
39	043D	20 E1		BRA	L1
40	043F	C1 81	NPLUS	CMPB	#\$81
41	0441	26 0D		BNE	NMINUS
42	0443	81 08		CMPA	#\$8
43	0445	27 D9		BEQ	L1
44	0447	C6 00		LDB	#\$00
45	0449	E7 86		STB	A,X
46	044B	4C		INCA	
47	044C	C6 00		LDB	#\$00
48	044E	20 E3		BRA	L2
49	0450	C1 84	NMINUS	CMPB	#\$84
50	0452	27 AC		BEQ	START
51	0454	C1 86		CMPB	#\$86
52	0456	26 C8		BNE	L1
53	0458	86 07		LDA	#\$7
54	045A	D6 10		LDB	MODE
55	045C	E7 86	L4	STB	A,X
56	045E	4A		DECA	
57	045F	2A FB		BPL	L4
58	0461	86 00		LDA	#\$0
59	0463	20 CE		BRA	L2
60	0465		END		

**Zu P5.3-9:**

1	F110	CLRDISP	EQU	\$F110
2	F133	SHOWDBUF	EQU	\$F133
3	F130	CLDBUF	EQU	\$F130
4	F140	KEY	EQU	\$F140
5	F160	DLY1MS	EQU	\$F160
6				
7				
8	0400	ORG		\$0400

9	0400	BD F1 10	START	JSR	CLRDISP
10	0403	BD F1 30		JSR	CLDBUF
11	0406	C6 01		LDB	#1
12	0408	86 07		LDA	#\$7
13	040A	E7 86		STB	A,X
14	040C	BD F1 33	L1	JSR	SHOWDBUF
15	040F	BD F1 40		JSR	KEY
16	0412	C1 FF		CMPB	#\$FF
17	0414	26 08		BNE	KEY1
18	0416	11 83 00 00		CMPU	#0
19	041A	27 F0		BEQ	L1
20	041C	20 04		BRA	L3
21	041E	C1 86	KEY1	CMPB	#\$86
22	0420	26 1D		BNE	NSTART
23	0422	10 8E 00 04	L3	LDY	#\$0004
24	0426	BD F1 60		JSR	DLY1MS
25	0429	E6 86		LDB	A,X
26	042B	58		LSLB	
27	042C	E7 86		STB	A,X
28	042E	C1 40		CMPB	#\$40
29	0430	25 DA		BLO	L1
30	0432	6F 86		CLR	A,X
31	0434	4A		DECA	
32	0435	2A 02		BPL	L2
33	0437	86 07		LDA	#\$7
34	0439	C6 01	L2	LDB	#1
35	043B	E7 86		STB	A,X
36	043D	20 CD		BRA	L1
37	043F				
38	043F	C1 80	NSTART	CMPB	#\$80
39	0441	26 05		BNE	NPLUS
40	0443	CE 00 01		LDU	#1
41	0446	20 C4		BRA	L1
42	0448	C1 81	NPLUS	CMPB	#\$81
43	044A	26 05		BNE	NMINUS
44	044C	CE 00 00		LDU	#0
45	044F	20 BB		BRA	L1
46	0451				
47	0451	C1 84	NMINUS	CMPB	#\$84
48	0453	26 B7		BNE	L1
49	0455	BD F1 30		JSR	CLDBUF
50	0458	CC 07 01		LDD	#\$0701
51	045B	E7 86		STB	A,X
52	045D	CE 00 00		LDU	#0
53	0460	20 AA		BRA	L1
54	0462		END		

**Zu P5.3-10:**

1	F110		CLRDISP	EQU	\$F110
2					
3	F140		KEY	EQU	\$f140
4	F143		HALTKEY	EQU	\$F143
5	F11C	SHOWT7SG		EQU	\$F11C
6	F120		SHOWB7SG	EQU	\$F120
7	F123		SHOWD7SG	EQU	\$F123
8	F156		SHOWADR	EQU	\$F156
9	F113		SHOWA	EQU	\$F113
10	F116		SHOWD	EQU	\$F116
11	F160		DLY1MS	EQU	\$F160
12	0000	HZ	EQU		\$0
13					
14	0400		ORG		\$0400
15	0400	BD F1 10	START	JSR	CLRDISP
16	0403	C6 87		LDB	#\$87
17	0405	20 39		BRA	L2
18	0407	BD F1 40	L1	JSR	KEY
19	040A	2B 34		BMI	L2
20					
21	040C	8E 00 00		LDX	#0
22	040F	BD F1 1C		JSR	SHOWT7SG
23					
24	0412	11 83 00 00		CMPU	#0
25	0416	26 04		BNE	L4
26	0418	C4 0C		ANDB	#\$0C ; "L"
27	041A	20 02		BRA	L5
28	041C	C4 03	L4	ANDB	#\$03 ; "S"
29					
30	041E	D7 00	L5	STB	\$0
31	0420	1F 98		TFR	B,A
32	0422	48		LSLA	
33	0423	48		LSLA	
34	0424	48		LSLA	
35	0425	48		LSLA	
36	0426	9A 00		ORA	\$0
37	0428	1F 89		TFR	A,B
38					
39	042A	11 83 00 00		CMPU	#0
40	042E	26 05		BNE	L6
41	0430	C3 01 23		ADDD	#\$0123 ; "L"
42	0433	20 03		BRA	L7
43	0435	C3 04 8C	L6	ADDD	#\$048C ; "S"



```

44
45 0438 8E 00 02          L7      LDX    #2
46 043B BD F1 23          JSR     SHOWD7SG
47 043E 20 C7            BRA     L1
48
49 0440 8E 00 07          L2      LDX    #7
50 0443 C1 86            CMPB   #$86
51 0445 26 07            BNE     L3
52 0447 CE FF FF          LDU     #$FFFF
53 044A 86 6D            LDA     #$6D    ;"S"
54 044C 20 09            BRA     MODE
55
56 044E C1 87            L3      CMPB   #$87
57 0450 26 0D            BNE     L9
58
59 0452 CE 00 00          LDU     #$0000
60 0455 86 38            LDA     #$38    ;"L"
61
62 0457 BD F1 10          MODE     JSR     CLRDISP
63 045A BD F1 13          JSR     SHOWA
64 045D 20 A8            BRA     L1
65 045F
66 045F C1 84            L9      CMPB   #$84
67 0461 26 A4            BNE     L1
68 0463 3F              SWI
69 0464                END

```

**Zu P5.3-11:**

```

1    F110      CLRDISP EQU      $F110
2
3    F140      KEY      EQU      $f140
4    F143      HALTKEY EQU      $F143
5    F11C              SHOWT7SG EQU  $F11C
6    F120      SHOWB7SG EQU      $F120
7    F123      SHOWD7SG EQU      $F123
8    F156      SHOWADR EQU      $F156
9    F113      SHOWA   EQU      $F113
10   F116              SHOWD   EQU  $F116
11   F160              DLY1MS  EQU  $F160
12   0000              HZ      EQU   $0
13
14 0400                ORG     $0400
15 0400 20 18      START  BRA     L12
16

```

17					
18	0402	BD F1 40	L1	JSR	KEY
19	0405	2A 1E		BPL	L10
20					
21	0407	C1 82		CMPB	#\$82 ; "A" = Koordinaten
22	0409	26 06	BNE	L11	
23	040B	5F	CLRB		
24	040C	CE 00 00		LDU	#0
25	040F	20 09		BRA	L12
26	0411	C1 83	L11	CMPB	#\$83 ; "D" = Quadranten
27	0413	26 ED		BNE	L1
28	0415	C6 01		LDB	#1
29	0417	CE 00 01		LDU	#1
30					
31	041A	8E 00 07	L12	LDX	#7
32	041D	BD F1 10		JSR	CLRDISP
33	0420	BD F1 1C		JSR	SHOWT7SG
34	0423	20 DD		BRA	L1
35					
36	0425	34 04 L10	PSHS	B	
37	0427	8E 00 00		LDX	#0
38	042A	BD F1 1C		JSR	SHOWT7SG
39					
40	042D	8E 00 03		LDX	#3
41	0430	11 83 00 00		CMPU	#0
42	0434	26 1C		BNE	L15
43	0436	86 1C		LDA	#\$1C ; "u"
44	0438	C4 08		ANDB	#\$08 ; "0,...,7" oder "8,...,F"
45	043A	27 02		BEQ	L51
46	043C	86 5C		LDA	#\$5C ; "o"
47	043E	BD F1 13	L51	JSR	SHOWA
48					
49	0441	30 1F	L5	LEAX	-1,X
50	0443	86 30	LDA	#\$38	;"I"
51	0445	35 04	PULS	B	
52	0447	C4 02		ANDB	#\$02
53	0449	27 02	BEQ	L4	
54	044B	86 50		LDA	#\$50 ; "r"
55	044D	BD F1 13	L4	JSR	SHOWA
56	0450	20 B0		BRA	L1
57					
58	0452	1F 98	L15	TFR	B,A
59	0454	84 03	ANDA	#3	
60	0456	97 00	STA	\$0	
61	0458	C4 0C		ANDB	#\$C
62	045A	58		LSLB	

63	045B	58		LSLB	
64	045C	DA 00		ORB	\$0
65	045E	8E 00 02		LDX	#2
66	0461	BD F1 20		JSR	SHOWB7SG
67					
68	0464	20 9C		BRA	L1
69					
70	0466			END	

**Zu P5.3-12:**

1	F110		CLRDISP	EQU	\$F110
2	F143		HALTKEY	EQU	\$F143
3	F140		KEY	EQU	\$F140
4	F120		SHOWB7SG	EQU	\$F120
5	F123		SHOWD7SG	EQU	\$F123
6	F156		SHOWADR	EQU	\$F156
7	F113		SHOWA	EQU	\$F113
8	F116		SHOWD	EQU	\$F116
9	F11C	SHOWT7SG		EQU	\$F11C
10	F160		DLY1MS	EQU	\$F160
11	0000	HZ0	EQU	\$0	
12	0001	HZ1	EQU	\$1	
13					
14	0400		ORG	\$0400	
15	0400	START			
16	0400	0F 00	L5	CLR	HZ0
17	0402	0F 01		CLR	HZ1
18	0404	10 8E 01 00		LDY	#\$0100
19	0408	BD F1 10		JSR	CLRDISP
20	040B	BD F1 43	L8	JSR	HALTKEY
21	040E	C1 86		CMPB	#\$86
22	0410	26 F9		BNE	L8
23	0412	BD F1 60		JSR	DLY1MS
24	0415	BD F1 10	L1	JSR	CLRDISP
25	0418	BD F1 40		JSR	KEY
26					
27	041B	C1 FF		CMPB	#\$FF
28	041D	27 F6		BEQ	L1
29	041F	8E 00 00		LDX	#\$0000
30	0422	2A 06		BPL	L4
31					
32	0424	86 0F		LDA	#\$F ; Funktionstaste
33	0426	0C 01		INC	HZ1
34	0428	20 04		BRA	L3
35	042A	86 0D	L4	LDA	#\$D ; Datentaste

36	042C	0C 00		INC	HZ0
37					
38	042E	C4 07	L3	ANDB	#\$07
39	0430	3A		ABX	
40	0431	1F 89		TFR	A,B
41	0433	BD F1 1C		JSR	SHOWT7SG
42	0436	36 04		PSHU	B
43	0438	BD F1 60		JSR	DLY1MS
44	043B	BD F1 40	L2	JSR	KEY
45	043E	C1 FF		CMPB	#\$FF
46	0440	26 F9		BNE	L2
47					
48	0442	DC 00		LDD	HZ0
49	0444	D1 00		CMPB	HZ0
50	0446	24 02		BHS	GREATER
51	0448	1E 89		EXG	A,B
52	044A	C1 0A	GREATER	CMPB	#\$A
53	044C	25 C7		BLO	L1
54					
55	044E	BD F1 10		JSR	CLRDISP
56	0451	8E 00 07		LDX	#7
57	0454	D6 00		LDB	HZ0
58	0456	BD F1 1C		JSR	SHOWT7SG
59	0459	8E 00 00		LDX	#0
60	045C	D6 01		LDB	HZ1
61	045E	BD F1 1C		JSR	SHOWT7SG
62	0461	BD F1 60		JSR	DLY1MS
63	0464	20 9A		BRA	L5
64	0466			END	

**Zu P5.3-13:**

1	F100	T7SG	EQU	\$F100
2	F110	CLRDISP	EQU	\$F110
3	F133	SHOWDBUF	EQU	\$F133
4	F139	RLDBUF	EQU	\$F139
5	F13C	COPYDBUF	EQU	\$F13C
6	F140	KEY	EQU	\$F140
7	F143	HALTKEY	EQU	\$F143
8	F11C	SHOWT7SG	EQU	\$F11C
9	F160	DLY1MS	EQU	\$F160
10				
11				
12	0400	ORG	\$0400	
13	0400	EINGABE JSR	CLRDISP	

14	0403	86 07		LDA	#\$7
15	0405	1F 89	L1	TFR	A,B
16	0407	1F 01		TFR	D,X
17	0409	BD F1 43		JSR	HALTKEY
18	040C	C1 0A		CMPB	#\$A
19	040E	24 F5		BHS	L1
20	0410	BD F1 1C		JSR	SHOWT7SG
21	0413	34 02		PSHS	A
22	0415	8E 00 00		LDX	#0
23	0418	BD F1 00		JSR	T7SG
24	041B	35 04		PULS	B
25	041D	A7 85		STA	B,X
26	041F	1F 98		TFR	B,A
27	0421	4A		DECA	
28	0422	2A E1		BPL	L1
29	0424	8E 00 10		LDX	#\$10
30	0427	10 8E 00 00		LDY	#\$00
31	042B	BD F1 3C		JSR	COPYDBUF
32	042E	1E 12		EXG	X,Y
33	0430	BD F1 33	WEITER	JSR	SHOWDBUF
34	0433	BD F1 43		JSR	HALTKEY
35	0436	C1 80		CMPB	#\$80
36	0438	26 05		BNE	NPLUS
37	043A	BD F1 39		JSR	RLDBUF
38	043D	20 F1		BRA	WEITER
39	043F	C1 84	NPLUS	CMPB	#\$84
40	0441	26 09		BNE	NRESET
41	0443	10 8E 00 10		LDY	#\$10
42	0447	BD F1 3C		JSR	COPYDBUF
43	044A	20 E4		BRA	WEITER
44	044C	C1 85	NRESET	CMPB	#\$85
45	044E	26 E0		BNE	WEITER
46	0450	10 8E 04 00		LDY	#\$400
47	0454	BD F1 60	L2	JSR	DLY1MS
48	0457	BD F1 39		JSR	RLDBUF
49	045A	BD F1 33		JSR	SHOWDBUF
50	045D	BD F1 40		JSR	KEY
51	0460	C1 86		CMPB	#\$86
52	0462	26 F0		BNE	L2
53	0464	20 CA		BRA	WEITER
54	0466		END		

**Zu P5.3-14:**

1	0400		ORG	\$0400
2	0400	BD F1 10	START	JSR CLRDISP
3	0403	8E 00 06		LDX #6
4	0406	C6 5A		LDB #\$5A
5	0408	BD F1 20		JSR SHOWB7SG
6	040B	8E 00 02		LDX #2
7	040E	BD F1 56		JSR SHOWADR
8	0411	10 9F 00		STY \$0
9	0414	8E 00 06		LDX #6
10	0417	C6 EA		LDB #\$EA
11	0419	BD F1 20		JSR SHOWB7SG
12	041C	8E 00 02		LDX #2
13	041F	BD F1 56		JSR SHOWADR
14	0422	BD F1 10		JSR CLRDISP
15	0425	1F 20		TFR Y,D
16	0427	93 00		SUBD \$0
17	0429	2A 06		BPL POSITIV
18	042B	81 FF		CMPA #\$FF
19	042D	27 0D		BEQ SHORT
20	042F	20 03		BRA LONG
21	0431	4D	POSITIV	TSTA
22	0432	27 08		BEQ SHORT
23	0434	BD F1 23	LONG	JSR SHOWD7SG
24	0437	CC 38 7C		LDD #\$387C
25	043A	20 06		BRA DARST
26	043C	BD F1 20	SHORT	JSR SHOWB7SG
27	043F	CC 6D 7C		LDD #\$6D7C
28	0442	8E 00 06	DARST	LDX #6
29	0445	BD F1 16		JSR SHOWD
30	0448	BD F1 43	TASTE	JSR HALTKEY
31	044B	C1 80		CMPB #\$80
32	044D	27 B1		BEQ START
33	044F	C1 81		CMPB #\$81
34	0451	26 F5		BNE TASTE
35	0453	8E 00 04		LDX #4
36	0456	DC 00		LDD \$0
37	0458	BD F1 23		JSR SHOWD7SG
38	045B	8E 00 00		LDX #0
39	045E	1F 20		TFR Y,D
40	0460	BD F1 23		JSR SHOWD7SG
41	0463	20 E3		BRA TASTE
42	0465			END
44	F110		CLRDISP	EQU \$F110

45	F143	HALTKEY	EQU	\$F143
46	F113	SHOWA	EQU	\$F113
47	F156	SHOWADR	EQU	\$F156
48	F116	SHOWD	EQU	\$F116
49	F120	SHOWB7SG	EQU	\$F120
50	F123	SHOWD7SG	EQU	\$F123
51				

**Zu P5.3-15:**

1	F110	CLRDISP	EQU	\$F110
2	F143	HALTKEY	EQU	\$F143
3	F120	SHOWB7SG	EQU	\$F120
4	F123	SHOWD7SG	EQU	\$F123
5	F156	SHOWADR	EQU	\$F156
6	F150	SHOWDATA	EQU	\$F150
7	F116	SHOWD	EQU	\$F116
8	F160	DLY1MS	EQU	\$F160
9	0000	E0	EQU	\$0
10	0001	E1	EQU	\$1
11				
12				
13	0400	ORG	\$0400	
14	0400 BD F1 10	EINGABE JSR	CLRDISP	
15	0403 8E 00 06	LDX	#\$0006	
16	0406 C6 E0	LDB	#\$E0	
17	0408 BD F1 20	JSR	SHOWB7SG	
18	040B 8E 00 04	LDX	#\$0004	
19	040E BD F1 50	JSR	SHOWDATA	
20	0411 97 00	STA	E0	
21	0413 8E 00 02	LDX	#\$0002	
22	0416 C6 E1	LDB	#\$E1	
23	0418 BD F1 20	JSR	SHOWB7SG	
24	041B 8E 00 00	LDX	#\$0000	
25	041E BD F1 50	JSR	SHOWDATA	
26	0421 97 01	STA	E1	
27				
28	0423 BD F1 10	START JSR	CLRDISP	
29	0426 CE 04 00	LDU	#\$0400	
30	0429 96 00	LOOP1 LDA	E0	
31	042B A1 C4	CMPA	,U	
32	042D 27 0B	BEQ	GEFUND1	
33	042F 33 41	LEAU	1,U	
34	0431 1F 30	TFR	U,D	
35	0433 5D	TSTB		

36	0434	26 F3		BNE	LOOP1
37	0436	D6 00		LDB	E0
38	0438	20 22		BRA	NICHT
39					
40	043A	8E 00 02	GEFUND1	LDX	#\$0002
41	043D	1F 30		TFR	U,D
42	043F	BD F1 23		JSR	SHOWD7SG
43	0442	8E 00 00		LDX	#\$0000
44	0445	E6 C0		LDB	,U+
45	0447	BD F1 20		JSR	SHOWB7SG
46	044A	10 8E 01 00		LDY	#\$0100
47	044E	BD F1 60		JSR	DLY1MS
48	0451	D1 01		CMPB	E1
49	0453	27 0D		BEQ	ENDE
50	0455	1F 30		TFR	U,D
51	0457	5D		TSTB	
52	0458	26 E0		BNE	GEFUND1
53	045A	D6 01		LDB	E1
54					
55	045C	BD F1 10	NICHT	JSR	CLRDISP
56	045F	BD F1 20		JSR	SHOWB7SG
57					
58	0462	13	ENDE	SYNC	
59	0463			END	

**Zu P5.3-16:**

1	F110		CLRDISP	EQU	\$F110
2	F140		KEY	EQU	\$F140
3	F143		HALTKEY	EQU	\$F143
4	F120		SHOWB7SG	EQU	\$F120
5	F123		SHOWD7SG	EQU	\$F123
6	F156		SHOWADR	EQU	\$F156
7	F150		SHOWDATA	EQU	\$F150
8	F116		SHOWD	EQU	\$F116
9	F160		DLY1MS	EQU	\$F160
10	0000	E0	EQU	\$0	
11	0001	E1	EQU	\$1	
12					
13					
14	0400		ORG	\$0400	
15	0400	BD F1 10	EINGABE	JSR	CLRDISP
16	0403	8E 00 06		LDX	#\$0006
17	0406	C6 E0		LDB	#\$E0
18	0408	BD F1 20		JSR	SHOWB7SG
19	040B	8E 00 00		LDX	#\$0000



20	040E	BD F1 50		JSR	SHOWDATA
21	0411	97 00		STA	E0
22	0413	8E 00 06		LDX	#\$0006
23	0416	C6 E1		LDB	#\$E1
24	0418	BD F1 20		JSR	SHOWB7SG
25	041B	8E 00 00		LDX	#\$0000
26	041E	BD F1 50		JSR	SHOWDATA
27	0421	97 01		STA	E1
28					
29	0423	CE 04 00	START	LDU	#\$0400
30	0426	BD F1 10	LOOP1	JSR	CLRDISP
31	0429	1F 30		TFR	U,D
32	042B	8E 00 02		LDX	#\$0002
33	042E	BD F1 23		JSR	SHOWD7SG
34	0431	E6 C0		LDB	,U+
35	0433	8E 00 00		LDX	#\$0000
36	0436	BD F1 20		JSR	SHOWB7SG
37	0439	10 8E 04 00		LDY	#\$0400
38					
39	043D	D1 00		CMPB	E0
40	043F	26 04		BNE	WEITER0
41	0441	C6 E0		LDB	#\$E0
42	0443	20 06		BRA	DISPLAY
43	0445	D1 01	WEITER0	CMPB	E1
44	0447	26 0C		BNE	WEITER
45	0449	C6 E1		LDB	#\$E1
46					
47	044B	8E 00 06	DISPLAY	LDX	#\$0006
48	044E	BD F1 20		JSR	SHOWB7SG
49	0451	10 8E 08 00		LDY	#\$0800
50					
51					
52	0455	BD F1 60	WEITER	JSR	DLY1MS
53	0458	BD F1 40		JSR	KEY
54	045B	C1 86		CMPB	#\$86
55	045D	27 05		BEQ	ENDE
56					
57	045F	1F 30		TFR	U,D
58	0461	5D		TSTB	
59	0462	26 C2		BNE	LOOP1
60	0464	13	ENDE	SYNC	
61	0465		END		

**Zu P5.3-17:**

1	0400	ORG	\$0400	
2	0400 BD F1 10	START	JSR	CLRDISP
8	0403 86 38		LDA	#\$38
9	0405 97 00		STA	\$0
10	0407 10 8E 00 C8		LDY	#0200
12	040B CE F0 00		LDU	#\$F000
13	040E 1F 30	TAKT	TFR	U,D
14	0410 1E 89		EXG	A,B
15	0412 1F 03		TFR	D,U
16	0414 B7 F0 02		STA	DRB
17	0417 BD F1 60		JSR	DLY1MS
25	041A F6 F0 02	SW1SW0	LDB	DRB
26	041D C4 03		ANDB	#3
27	041F 8E 00 00		LDX	#0
28	0422 BD F1 1C		JSR	SHOWT7SG
29	0425 86 08		LDA	#8
30	0427 48	SHIFT	LSLA	
31	0428 5A		DECB	
32	0429 2A FC		BPL	SHIFT
33				
34	042B BD F1 40	CB2	JSR	KEY
35	042E C1 80		CMPB	#\$80
36	0430 26 04		BNE	MINUS
37	0432 C6 38		LDB	#\$38
38	0434 20 06		BRA	WEITER
39	0436 C1 81	MINUS	CMPB	#\$81
40	0438 26 04		BNE	L2
41	043A C6 30		LDB	#\$30
42	043C D7 00	WEITER	STB	\$0
43	043E C6 00	L2	LDB	#0
44	0440 F7 F0 03		STB	CRB
45	0443 B7 F0 02		STA	DDRB
46	0446 C6 04		LDB	#4
47	0448 DA 00		ORB	\$0
48	044A F7 F0 03		STB	CRB
49	044D 20 BF		BRA	TAKT
50			END	
51	F110	CLRDISP	EQU	\$F110
52	F11C	SHOWT7SG	EQU	\$F11C
53	F140	KEY	EQU	\$F140
54	F002	DRB	EQU	\$F002
55	F002	DDRB	EQU	\$F002
56	F003	CRB	EQU	\$F003
57	F160	DLY1MS	EQU	\$F160

**Zu P5.3-18:**

1	F110		CLRDISP	EQU	\$F110
2	F140		KEY	EQU	\$F140
3	F143		HALTKEY	EQU	\$F143
4	F113		SHOWA	EQU	\$F113
5	F11C	SHOWT7SG		EQU	\$F11C
6	F120		SHOWB7SG	EQU	\$F120
7	F123		SHOWD7SG	EQU	\$F123
8	F156		SHOWADR	EQU	\$F156
9	F150		SHOWDATA	EQU	\$F150
10	F116		SHOWD	EQU	\$F116
11	F160		DLY1MS	EQU	\$F160
12	0000	E0	EQU	\$0	
13	0001	E1	EQU	\$1	
14	F002		DRB	EQU	\$F002
15	F002		DRRB	EQU	\$F002
16	F003		CRB	EQU	\$F003
17					
18					
19	0400		ORG	\$0400	
20	0400	BD F1 10	EINGABE	JSR	CLRDISP
21	0403	C6 30		LDB	#\$30
22	0405	F7 F0 03		STB	CRB
23	0408	C6 FF		LDB	#\$FF
24	040A	F7 F0 02		STB	DRRB
25	040D	C6 34		LDB	#\$34
26	040F	F7 F0 03		STB	CRB
27	0412	8E 00 07	LOOP	LDX	#\$0007
28					
29	0415	BD F1 40		JSR	KEY
30	0418	2B F8		BMI	LOOP
31	041A	BD F1 1C		JSR	SHOWT7SG
32					
33	041D	CB 30	ASCII	ADDB	#\$30
34	041F	C1 3A		CMPB	#\$3A
35	0421	2B 02		BMI	ZIFF
36	0423	CB 07		ADDB	#\$07
37	0425	8E 00 04	ZIFF	LDX	#\$0004
38	0428	BD F1 20		JSR	SHOWB7SG
39	042B	B6 F0 03		LDA	CRB
40	042E	2A E2		BPL	LOOP
41	0430	F7 F0 02		STB	DRB
42	0433	F6 F0 02		LDB	DRB
43					
44					

45	0436	86 3C	LDA	#\$3C
46	0438	B7 F0 03	STA	CRB
47	043B	8E 00 00	LDX	#0
48	043E	C6 00	LDB	#0
49	0440	BD F1 1C	JSR	SHOWT7SG
50	0443	10 8E 01 00	LDY	#\$0100
51	0447	BD F1 60	JSR	DLY1MS
52	044A	86 34	LDA	#\$34
53	044C	B7 F0 03	STA	CRB
54	044F	86 00	LDA	#0
55	0451	BD F1 13	JSR	SHOWA
56	0454	20 BC	BRA	LOOP
57	0456		END	

**Zu P5.3-19:**

1	F110		CLRDISP	EQU	\$F110
2	F140		KEY	EQU	\$F140
3	F143		HALTKEY	EQU	\$F143
4	F113		SHOWA	EQU	\$F113
5	F116		SHOWD	EQU	\$F116
6	F11C	SHOWT7SG		EQU	\$F11C
7	F120		SHOWB7SG	EQU	\$F120
8	F123		SHOWD7SG	EQU	\$F123
9	F156		SHOWADR	EQU	\$F156
10	F150		SHOWDATA	EQU	\$F150
11	F160		DLY1MS	EQU	\$F160
12	0000	E0	EQU	\$0	
13	0001	E1	EQU	\$1	
14	F002		DRB	EQU	\$F002
15	F002		DRRB	EQU	\$F002
16	F003		CRB	EQU	\$F003
17					
18					
19	0400		ORG	\$0400	
20	0400	BD F1 10	EINGABE	JSR	CLRDISP
21	0403	C6 30		LDB	#\$30
22	0405	F7 F0 03		STB	CRB
23	0408	C6 00		LDB	#\$00
24	040A	F7 F0 02		STB	DRRB
25	040D	C6 34		LDB	#\$34
26	040F	F7 F0 03		STB	CRB
27					
28	0412	8E 00 00	LOOP	LDX	#0
29	0415	BD F1 40		JSR	KEY
30	0418	C1 80		CMPB	#\$80

31	041A	26 0C		BNE	MINUS
32					
33	041C	C6 3C	PLUS	LDB	#\$3C
34	041E	F7 F0 03		STB	CRB
35	0421	C6 01		LDB	#\$1
36	0423	BD F1 1C		JSR	SHOWT7SG
37	0426	20 0E		BRA	WEITER
38					
39	0428	C1 81	MINUS	CMPB	#\$81
40	042A	26 0A		BNE	WEITER
41	042C	C6 34		LDB	#\$34
42	042E	F7 F0 03		STB	CRB
43	0431	C6 00		LDB	#0
44	0433	BD F1 1C		JSR	SHOWT7SG
45					
46	0436	8E 00 06	WEITER	LDX	#6
47	0439	F6 F0 03		LDB	CRB
48	043C	C4 08		ANDB	#\$08
49	043E	26 0B		BNE	ENABLE
50					
51	0440	F6 F0 02	DISABLE	LDB	DRB
52	0443	CC 00 00		LDD	#0
53	0446	BD F1 16		JSR	SHOWD
54	0449	20 C7		BRA	LOOP
55					
56	044B	F6 F0 03	ENABLE	LDB	CRB
57	044E	2A C2		BPL	LOOP
58	0450	F6 F0 02		LDB	DRB
59	0453	BD F1 20		JSR	SHOWB7SG
60	0456	20 BA		BRA	LOOP
61	0458		END		

**Zu P5.3-20:**

```

1      002E      INTZ3      EQU      $002E
2      F018      SRZ1       EQU      $F018
3      F018      SRZ3       EQU      $F018
4      F019      SRZ2       EQU      $F019
5      F019      STREG      EQU      $F019
6      F110      CLRDISP    EQU      $F110
7      F01E      LATCHZ3    EQU      $F01E
8      F116      SHOWA      EQU      $F116
9      F123      SHOWD7SG   EQU      $F123
10     F140      KEY         EQU      $F140
11     F143      HALTKEY     EQU      $F143
12
13     0400                      ORG    $0400
14     0400 BD F1 10      START JSR    CLRDISP
15     0403 8E 00 02                      LDX    #$0002
16     0406 BD F1 23                      JSR    SHOWD7SG
17     0409 CC 04 41                      LDD    #INT
18     040C DD 2E                      STD    INTZ3
19     040E 7F F0 19                      CLR    SRZ2
20     0411 86 C3                      LDA    #$E3
21     0413 B7 F0 18                      STA    SRZ3
22     0416 CC 01 FF                      LDD    #$01FF
23     0419 FD F0 1E                      STD    LATCHZ3
24     041C 86 01                      LDA    #$01
25     041E B7 F0 19                      STA    SRZ2
26     0421 BD F1 43      WDTS   JSR    HALTKEY
27     0424 C1 86                      CMPB   #$86
28     0426 26 F9                      BNE    WDTS
29
30     0428 7F F0 18                      CLR    SRZ1
31
32     042B 1F 20                      LOOP  TFR    Y,D
33     042D BD F1 23                      JSR    SHOWD7SG
34     0430 31 21                      LEAY   1,Y
35     0432 BD F1 40                      JSR    KEY
36     0435 C1 84                      CMPB   #$84
37     0437 26 F2                      BNE    LOOP
38     0439 CC 01 FF                      LDD    #$01FF
39     043C FD F0 1E                      STD    LATCHZ3
40     043F 20 EA                      BRA    LOOP
41     0441
42     0441 86 01                      INT    LDA    #$01
43     0443 B7 F0 18                      STA    SRZ1
44     0446 8E 00 02                      LDX    #$0002
45     0449 86 80                      LDA    #$80
46     044B BD F1 16                      JSR    SHOWA
47     044E BD F1 43      WAIT   JSR    HALTKEY
48     0451 C1 86                      CMPB   #$86
49     0453 26 F9                      BNE    WAIT
50     0455 7F F0 18                      CLR    SRZ1
51     0458 F6 F0 19                      LDB    STREG
52     045B FC F0 1E                      LDD    LATCHZ3
53     045E CC 01 FF                      LDD    #$01FF
54     0461 FD F0 1E                      STD    LATCHZ3
55
56     0464 3B                      RTI

```

**Zu P5.3-21:**

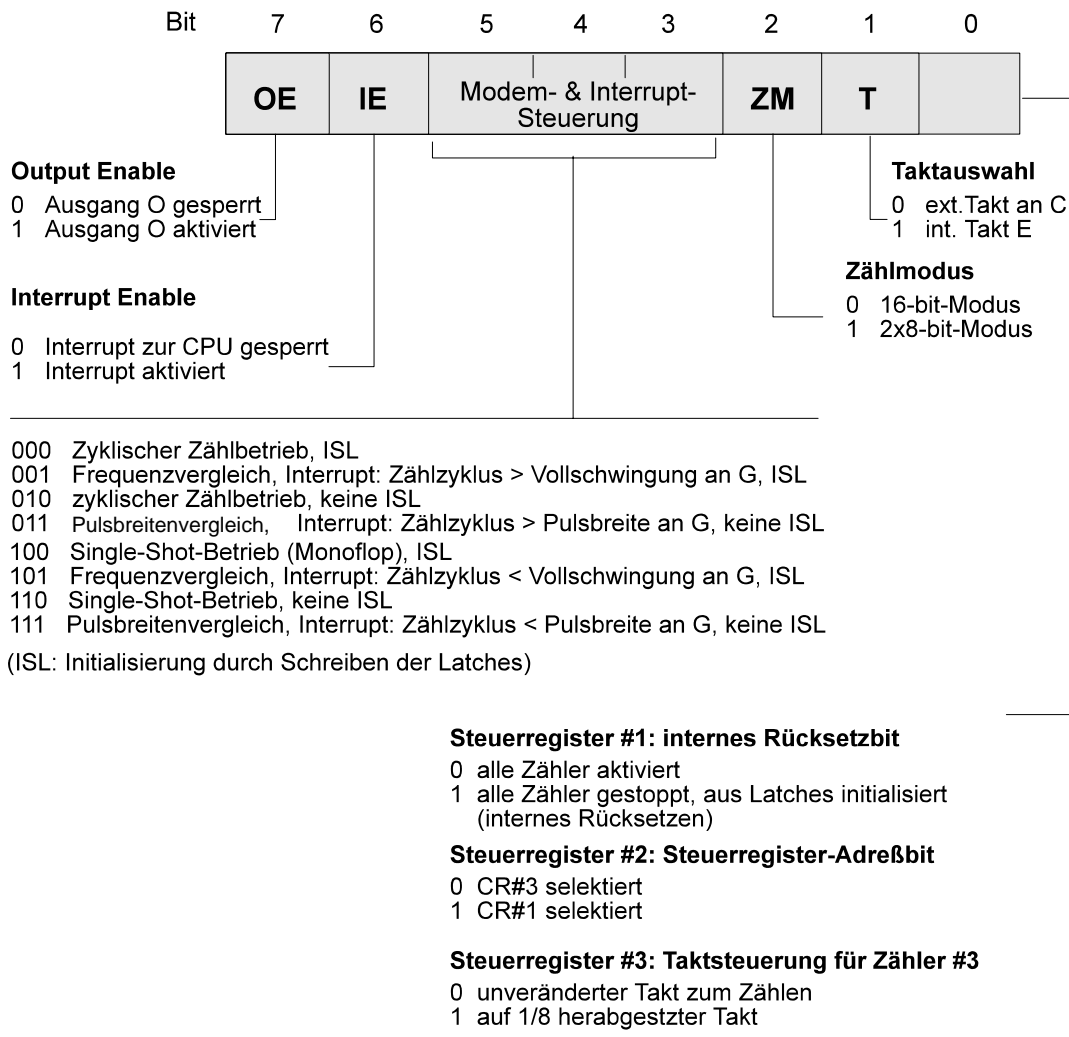
```

1      002E      INTZ3      EQU    $002E
2      F018      SRZ1EQU    $F018
3      F018      SRZ3EQU    $F018
4      F019      SRZ2EQU    $F019
5      F019      STREG      EQU    $F019
6      F110      CLRDISP    EQU    $F110
7      F01E      LATCH_Z3   EQU    $F01E
8      F11C      SHOWT7SG   EQU    $F11C
9      F123      SHOWD7SG   EQU    $F123
10     F143      HALTKEY     EQU    $F143
11     F160      DLY1MS      EQU    $F160
12
13 0400                      ORG    $0400
14 0400 CC 04 52      START LDD    #INT
15 0403 DD 2E                      STD    INTZ3
16 0405 7F F0 19                      CLR    SRZ2
17 0408 86 C3                      LDA    #$C3
18 040A B7 F0 18                      STA    SRZ3
19 040D CC 80 FF                      LDD    #$80FF
20 0410 FD F0 1E                      STD    LATCHZ3
21 0413 86 01                      LDA    #$01
22 0415 B7 F0 19                      STA    SRZ2
23
24 0418 BD F1 10  ANFANG JSR    CLRDISP
25 041B 8E 00 02                      LDX    #$0002
26 041E CC 00 00                      LDD    #$0000
27 0421 DD 00                      STD    $00
28 0423 BD F1 23                      JSR    SHOWD7SG
29 0426 30 1E                      LEAX   -2,X
30 0428 C6 05                      LDB    #$05
31 042A BD F1 1C                      JSR    SHOWT7SG
32
33 042D BD F1 43  WDTS JSR    HALTKEY
34 0430 C1 86                      CMPB   #$85
35 0432 26 F9                      BNE    WDTS
36
37 0434 7F F0 18                      CLR    SRZ1
38 0437 C6 0C                      LDB    #$0C
39 0439 BD F1 1C                      JSR    SHOWT7SG
40 043C
41 043C C6 01      LOOP LDB    #$01
42 043E D7 02                      STB    $02
43 0440 10 8E 40 00 LDY    #$4000
44 0444 BD F1 60                      JSR    DLY1MS
45 0447 D6 02                      LDB    $02
46 0449 27 F1                      BEQ    LOOP
47 044B 86 01                      LDA    #$01
48 044D B7 F0 18                      STA    SRZ1
49 0450 20 C6                      BRA    ANFANG
50 0452
51 0452 0F 02      INT CLR    $02
52 0454 DC 00                      LDD    $00
53 0456 C3 00 01 ADDD   #$0001
54 0459 DD 00                      STD    $00
55 045B 8E 00 02 LDX    #$0002
56 045E BD F1 23 JSR    SHOWD7SG
57 0461 F6 F0 19 LDB    STREG
58 0464 FC F0 1E LDD    LATCHZ3
59 0467 3B                      RTI

```

## Anhang Der Registersatz des MC6840

### Steuerregister CR#1, CR#2, CR#3:



### Registeradressen:

CR#1: \$F018 und Bit 0 von CR#2=1  
 CR#2: \$F019  
 CR#3: \$F018 und Bit 0 von CR#2=0

Latches: \$F01A, \$F01B  
 Latches: \$F01C, \$F01D  
 Latches: \$F01E, \$F01F

### Statusregister:

(Adresse: \$F019)

