

MFA-MEDIENSYSTEM

# Mikrocomputer- Technik

Handbuch zum  
Leybold-Ampelmodell

  
MEDIENSYSTEM

vgs



# **Steuerung des Verkehrsampel-Modells mit dem MFA-Mikrocomputer-System**

Ein Kurs zur Steuerung von Anlagen







**Steuerung des Verkehrsampel-Modells  
mit dem MFA-Mikrocomputer-System**

**ISBN 3-8025-1255-3**

**© LEYBOLD DIDACTIC GmbH  
Leybold-Straße 1  
5030 Hürth**

**© vgs Verlagsgesellschaft mbH & Co.KG  
Breite Straße 118/120  
5000 Köln 1**

**Text und Programmierung:  
Dr. Roland Schulé  
Software & Dokumentation  
Im Martinskirchle 8  
7244 Waldachtal 3**

**Das Werk und seine Teile sind urheberrechtlich geschützt. Jede Verwertung in anderen als den gesetzlich zugelassenen Fällen bedarf deshalb der vorherigen schriftlichen Einwilligung der LEYBOLD DIDACTIC GmbH oder der vgs Verlagsgesellschaft mbH & Co.KG.**

**Die als Aufgabenblätter oder Kopiervorlagen gekennzeichneten Seiten dürfen in der für den Unterricht erforderlichen Anzahl kopiert werden.**

1.	Vorwort	5
2.	Beschreibung des Modells	7
2.1	Die Situation in der Realität	7
2.2	Vereinfachungen im Modell	8
2.3	Das Technologieschema	9
2.4	Die Verkehrsampeln	11
2.5	Die Fußgängerampeln	12
2.6	Die Melde-Einrichtungen	13
2.7	Die Schaltuhr	13
2.8	Die Funktionsbeschreibung	14
3.	Der Anschluß an den MFA-Mikrocomputer	17
3.1	Liste der erforderlichen Baugruppen	17
3.2	Modifikation des MFA-Mikrocomputers	20
3.3	Die Zuordnungsliste	22
3.4	Anschluß an die E/A-Baugruppen	24
3.5	Stromversorgung des Verkehrsampel-Modells	25
3.6	Programme auf der Diskette im BFZ-Mini-DOS-Format	26
3.7	Programme auf der Diskette im BFZ-CP/M-Format	29
4.	Aufgaben	30
4.1	Aufgabe A1: Aufbau des Modells	30
4.2	Aufgabe A2: Erkundung der Verkehrsampeln	32
4.3	Aufgabe A3: Erkundung der Fußgängerampeln	34
4.4	Aufgabe A4: Erkundung der Melde-Einrichtungen	36
4.5	Aufgabe A5: Steuerung der Verkehrsampeln	38
4.5.1	Steuerung der Verkehrsampeln (Assembler)	40
4.5.2	Steuerung mit dem Tabellenverfahren (Assembler)	44
4.5.3	Steuerung der Verkehrsampeln (BASIC)	48
4.5.4	Steuerung mit dem Tabellenverfahren (BASIC)	52
4.5.5	Steuerung der Verkehrsampeln (SPS)	54
4.6	Aufgabe A6: Steuerung der Verkehrs- und Fußgängerampeln	60
4.6.1	Steuerung aller Ampeln (Assembler)	62
4.6.2	Erweiterung der Steuerungstabelle (Assembler)	66
4.6.3	Steuerung aller Ampeln (BASIC)	70
4.6.4	Erweiterung der Steuerungstabelle (BASIC)	72
4.6.5	Steuerung aller Ampeln (SPS)	74
4.7	Aufgabe A7: Bedarfssteuerung der Nebenstraße	80
4.7.1	Steuerungsprogramm für die Bedarfssteuerung (Assembler)	82
4.7.2	" (BASIC)	86
4.7.3	" (SPS)	90
4.8	Aufgabe A8: Bedarfssteuerung der Nebenstraße und des Fußwegs	94
4.8.1	Erweiterte Bedarfssteuerung (Assembler)	100
4.8.2	" (BASIC)	106
4.8.3	" (SPS)	110
4.9	Aufgabe A9: Speicherung der Melde-Impulse	120
4.9.1	" (Assembler)	122
4.9.2	" (BASIC)	128
4.9.3	" (SPS)	132

---

## Inhaltsverzeichnis

---

4.10	Aufgabe A10: Verkehrsdichten-abhängige Steuerung	140
4.10.1	Steuerungsprogramm mit Berücksichtigung der Verkehrsdichte	
	(Assembler)	144
4.10.2	"	(BASIC) 158
4.10.3	"	(SPS) 164
4.11	Aufgabe A11: Tag/Nacht-Umschaltung	176
4.11.1	Betriebsarten-Umschaltung	(Assembler) 178
4.11.2	"	(BASIC) 194
4.11.3	"	(SPS) 200
5.	Lösungen	214
5.1	Aufgabe A1: Aufbau des Modells	214
5.2	Aufgabe A2: Erkundung der Verkehrsampeln	215
5.3	Aufgabe A3: Erkundung der Fußgängerampeln	216
5.4	Aufgabe A4: Erkundung der Melde-Einrichtungen	217
5.5	Aufgabe A5: Steuerung der Verkehrsampeln	218
5.5.1	Steuerung der Verkehrsampeln (Assembler)	219
5.5.2	Steuerung mit dem Tabellenverfahren (Assembler)	220
5.5.3	Steuerung der Verkehrsampeln (BASIC)	221
5.5.4	Steuerung mit dem Tabellenverfahren (BASIC)	222
5.5.5	Steuerung der Verkehrsampeln (SPS)	223
5.6	Aufgabe A6: Steuerung der Verkehrs- und Fußgängerampeln	224
5.6.1	Steuerung aller Ampeln (Assembler)	225
5.6.2	Erweiterung der Steuerungstabelle (Assembler)	226
5.6.3	Steuerung aller Ampeln (BASIC)	227
5.6.4	Erweiterung der Steuerungstabelle (BASIC)	228
5.6.5	Steuerung der Ampeln (SPS)	229
5.7	Aufgabe A7: Bedarfssteuerung der Nebenstraße	231
5.7.1	Steuerungsprogramm für die Bedarfssteuerung	
	(Assembler, BASIC und SPS)	232
5.8	Aufgabe A8: Bedarfssteuerung der Nebenstraße und des Fußwegs	233
5.8.1	Erweiterte Bedarfssteuerung (Assembler, BASIC und SPS)	236
5.9	Aufgabe A9: Speicherung der Melde-Impulse	237
5.9.1	"	(Assembler) 238
5.9.2	"	(BASIC) 240
5.9.3	"	(SPS) 241
5.10	Aufgabe A10: Verkehrsdichten-abhängige Steuerung	242
5.10.1	Steuerungsprogramm (Assembler, BASIC und SPS)	244
5.11	Aufgabe A11: Tag/Nacht-Umschaltung	245
5.11.1	Betriebsarten-Umschaltung	(Assembler) 246
5.11.2	"	(BASIC) 247
5.11.3	"	(SPS) 248
6.	Kopiervorlagen für Projektionsfolien	250
6.1	Realität und Technologieschema	250
6.2	Ansicht des Schalttafelmodells VERKEHRSAMPEL	251
6.3	Zuordnungsliste	252
6.4	Schaltbild der elektrischen Anschlüsse	253
6.5	Zeitgeführter und prozeßgeführter Schritt	(Signal-Zeit-Diagramme) 254
6.6	"	(Programmablaufplan) 255
6.7	"	(Funktionsplan) 256
6.8	Programmverzweigung (Programmablaufplan und Funktionsplan)	257
6.9	Flankensteuerung (Signal-Zeit-Diagramm und Zahlendarstellung)	258

---

## **1. Vorwort**

Das vorliegende Handbuch für Ausbilder beschreibt den Betrieb eines Verkehrsampel-Modells am MFA-Mikrocomputer.

Mit dem Verkehrsampel-Modell wird das Einsatzgebiet des MFA-Mikrocomputers erweitert. Nicht mehr der Computer selbst ist Objekt des Unterrichts, sondern sein Einsatz in der Steuerung einer Anlage.

Bei dem Verkehrsampel-Modell handelt es sich um eine Experimentierplatte, die die Einmündung einer Seitenstraße in eine Hauptstraße zeigt. Um die Situation nicht unübersichtlich werden zu lassen, wird sowohl von der Hauptstraße als auch der Nebenstraße angenommen, daß es sich um Einbahnstraßen handelt. An der Nebenstraße befindet sich ein Überweg für Fußgänger, an der Hauptstraße deren zwei. Das Verkehrsaufkommen wird durch das Betätigen von Tastern dem Steuerungsrechner gemeldet; im Falle der Fahrbahnen stellen die Taster eine Simulation der üblicherweise installierten Induktionsschleifen dar.

Die Aufgabenstellung lautet, die Verkehrs- und Fußgängerampeln so zu programmieren, daß keine Kollisionen auftreten und der Verkehr je nach Aufkommen optimal fließen kann.

Ein wichtiger Bestandteil einer jeder Anlage ist die technische Dokumentation, wie sie in *Kapitel 2* enthalten ist. Sie umfaßt die Beschreibung der Signalgeber und der Stellglieder, das Technologieschema, die Funktionsbeschreibung und die Zuordnungsliste. Die technische Dokumentation sollte sorgfältig diskutiert werden und anschließend bei allen Programmierarbeiten griffbereit sein, um den gedanklichen Zusammenhang zwischen den Programmstrukturen und den Betriebsmitteln einer Anlage herzustellen.

Der Anschluß an den MFA-Mikrocomputer erfolgt mittels zweier Parallel-Schnittstellen, je einer für die Eingabe und die Ausgabe der Steuersignale. Die weitere Ausstattung des MFA-Mikrocomputers wird entsprechend der Programmiermethode variieren. *Kapitel 3* stellt die Varianten gegenüber. Es enthält zudem die Inhaltsangabe der beigefügten Disketten.

Die Aufgaben (*Kapitel 4*) gliedern sich in zwei Abschnitte. Zunächst werden auf theoretischer und auf experimenteller Basis die Ein- und Ausgänge des Steuerungscomputers und die Reaktionen des Verkehrsampel-Modells erkundet. Dieser Schritt erfolgt mit dem Technologieschema, der Zuordnungsliste, usw. sowie den Betriebssystem-Kommandos des MAT-85.

Der zweite Abschnitt betrifft die Programmierung. Je nach verwendeter Programmiersprache (Assembler, BASIC und SPS) trennen sich die Wege; zu unterschiedlich sind die Möglichkeiten und damit die Vorgehensweise. Die generelle Methode bleibt jedoch die gleiche. In der Erarbeitung des Steuerungsprogramms des Verkehrsampel-Modells wird mit Programm-Modulen gearbeitet, die einzeln getestet und zu Programmen wachsender Komplexität zusammengesetzt werden. Diese Vorgehensweise mag zeit-

---

## 1. Vorwort

---

aufwendig erscheinen. Die "verdaubaren" Einzelschritte führen jedoch zu frühen Erfolgserlebnissen, und der Lernende kommt zielstrebig zur Lösung der Gesamtaufgabe. Hätte die abschließende Lösung "aus dem Stand" erarbeitet werden müssen, wäre sie vielleicht wegen der Komplexität der Aufgabenstellung überhaupt nie zustande gekommen.

Jede Aufgabe ist wiederum unterteilt in einen Informationsteil (jeweils auf der linken Seite) und die Aufgabenblätter (jeweils auf der rechten Seite). Der Informationsteil liefert die notwendigen Informationen zur Bearbeitung der Arbeitsblätter und ist von seiner Diktion her eher für den Ausbilder gedacht. Es spricht dennoch nichts dagegen, auch diesen Teil dem Kursteilnehmer an die Hand zu geben, denn die Lösungen der Aufgaben werden nicht "verraten".

Anders ist es bei den aufeinanderfolgenden Aufgabenblättern. Da die Aufgaben, wie zuvor erwähnt, aufeinander aufbauen und oft aus Programm-ergänzungen bestehen, enthält die nächste Aufgabenstellung zum Teil die Lösung der jetzigen. In diesen Fällen wird es sinnvoll sein, die nächste Aufgabe erst nach Bearbeitung der vorliegenden auszuteilen.

Einige Aufgaben sind mit dem Zeichen ▲ gekennzeichnet und stellen Erweiterungen dar. Die Lösung der Erweiterungen wird nicht unmittelbar für die Bearbeitung des nachfolgenden Stoffs benötigt. Ihre Lösung bietet sich an, wenn man die Lerninhalte vertiefen will und ausreichend Zeit zur Verfügung hat, insbesondere bei leistungsstarken Kursen.

Der Kurs muß auch nicht vollständig durchgearbeitet werden. Je nach den zu vermittelnden Kenntnissen kann der Kurs nur bestimmte Abschnitten umfassen:

- |                                  |                     |
|----------------------------------|---------------------|
| - Ein- und Ausgabe, Schrittkette | - Aufgaben 1 bis 6, |
| - Prozeßgeführte Schritte        | - bis Aufgabe 7,    |
| - Programmverzweigungen          | - bis Aufgabe 8,    |
| - Flankensteuerung               | - bis Aufgabe 9,    |
| - Test komplexer Programme       | - bis Aufgabe 10,   |
| - Betriebsartenumschaltung       | - alle Aufgaben.    |

Die Aufgabenblätter des *Kapitels 4* können in der kursüblichen Anzahl fotokopiert und dem Kursteilnehmer ausgehändigt werden. Die Aufgabenblätter sollten ausgefüllt und ggf. mit weiteren Unterlagen, wie Beschreibungen oder Programmlisten, ergänzt werden.

Das *Kapitel 5* resümiert kurz die Aufgabenstellungen und gibt die Lösungen (bei Programmen für alle drei Programmiersprachen) wieder. Bei den Programmen wird kein Anspruch erhoben, daß die angegebenen Lösungen die allein richtigen Möglichkeiten darstellen; Varianten mögen ebenfalls zum Ziel führen.

Der Band schließt mit Kopiervorlagen für Projektionsfolien in *Kapitel 6*.

## 2. Beschreibung des Modells

### 2.1 Die Situation in der Realität

Bei der Funktionsbeschreibung des Verkehrsampel-Modells wird von einer Straßeneinmündung mit Ampelregelung ausgegangen. Es gibt eine Hauptstraße, erkennbar an der größeren Straßenbreite, sowie eine Nebenstraße, die unter spitzem Winkel von links hinzukommt. Beide Straßen sind Einbahnstraßen; die Verkehrsflüsse vereinigen sich in der Hauptstraße. Dies stellt eine recht einfache Situation dar, denn für jede Fahrtrichtung ist der weitere Fahrtverlauf eindeutig gegeben. Eine Aufspaltung des Verkehrsflusses in mehrere Richtungen oder Gegenverkehr müssen nicht beachtet werden.

Der Einmündungsbereich der Straßen ist mit insgesamt drei Fußgängerüberwegen ausgestattet, die ebenfalls Ampelregelung besitzen. Je ein Überweg befindet sich auf der Hauptstraße vor und nach der Einmündung, ein dritter verläuft über die Nebenstraße.

An beiden Straßen befinden sich vor dem Fußgängerüberweg und damit auch vor der Einmündung Verkehrsampeln, die, wie allgemein üblich, jeweils eine rote, gelbe und grüne Lampe besitzen.

Die Fußgängerüberwege sind ebenfalls mit Ampeln ausgestattet. Diese Ampeln besitzen jeweils nur eine rote und eine grüne Lampe.

In der Fahrbahn der beiden einmündenden Straßen sind jeweils Induktionsschleifen verlegt. Diese reagieren auf Metall, d.h. auf ein Fahrzeug, mit einem Schaltimpuls.

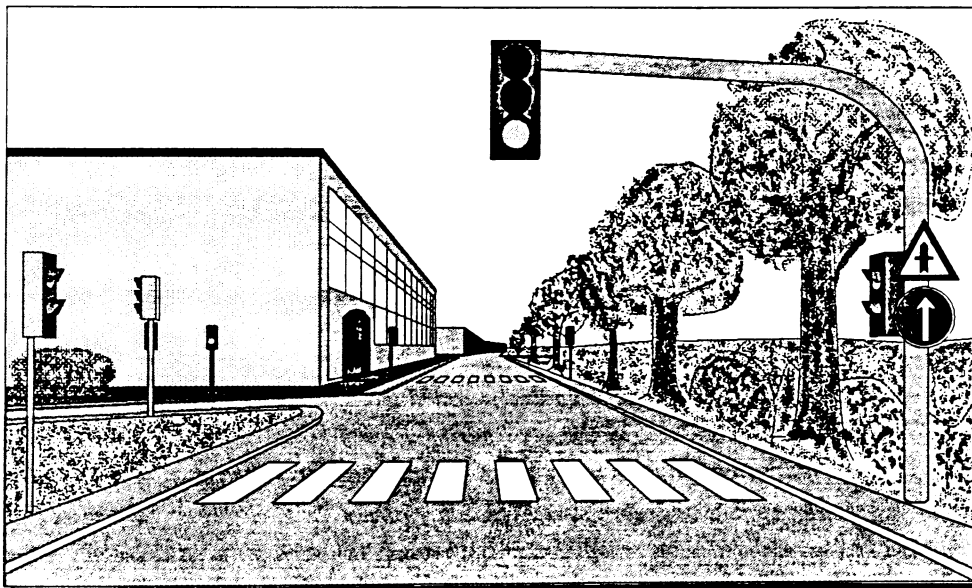


Abbildung 2.1: Ansicht des Einmündungsbereichs

---

## 2. Beschreibung des Modells

---

Der Fußgängerüberweg nach der Zusammenführung der beiden Straßen ist auf jeder Straßenseite mit einer Drucktaste ausgestattet. Durch Drücken der Taste meldet ein Fußgänger seinen Wunsch an, die Straße zu überqueren. Die beiden Fußgängerüberwege vor der Einmündung besitzen eine solche Meldeeinrichtung nicht.

Die Ampelanlage wird durch einen Verkehrsrechner gesteuert. In der Praxis mag er anders aussehen als das im Kurs verwendete MFA-Mikrocomputer-System; funktionell ergeben sich jedoch kaum Unterschiede.

### 2.2 Vereinfachungen im Modell

Im Unterrichtseinsatz müssen Abstriche an dem zuvor gezeichneten Bild einer Straßeneinmündung vorgenommen werden. So ist es weder möglich noch nötig, reale Verkehrsampeln zu steuern. Sie können ohne funktionelle Abstriche durch die roten, gelben und grünen Leuchtdioden der Experimentierplatte LEYBOLD DIDACTIC 72856 ersetzt werden.

Die Fußgängerampeln sind auf der Experimentierplatte jeweils nur einfach angebracht. In Realität wäre natürlich jeweils ein Paar von Ampeln installiert, für jede Gehrichtung eine.

Das gleiche gilt für die Meldetasten; hier genügt ebenfalls eine Taste. Das Betätigen der Taste stellt einen Tastendruck auf der einen *oder* der anderen Seite in Realität dar.

Die Induktionsschleifen in der Fahrbahn der beiden Straßen sind durch Tasten ersetzt. Ein Tastendruck simuliert hier ein Fahrzeug, das sich über der Induktionsschleife befindet.

Im Gegensatz zu dem Modell ist der MFA-Mikrocomputer keine "vereinfachte" Ausführung eines Steuerungscomputers, sondern ein vollwertiger Computer. Es ist wichtig, dem Kursteilnehmer bewußt zu machen, daß die Arbeit mit dem Computer und das daraus folgende Steuerungsprogramm sich keineswegs von der Situation bei einer realen Anlage unterscheiden.

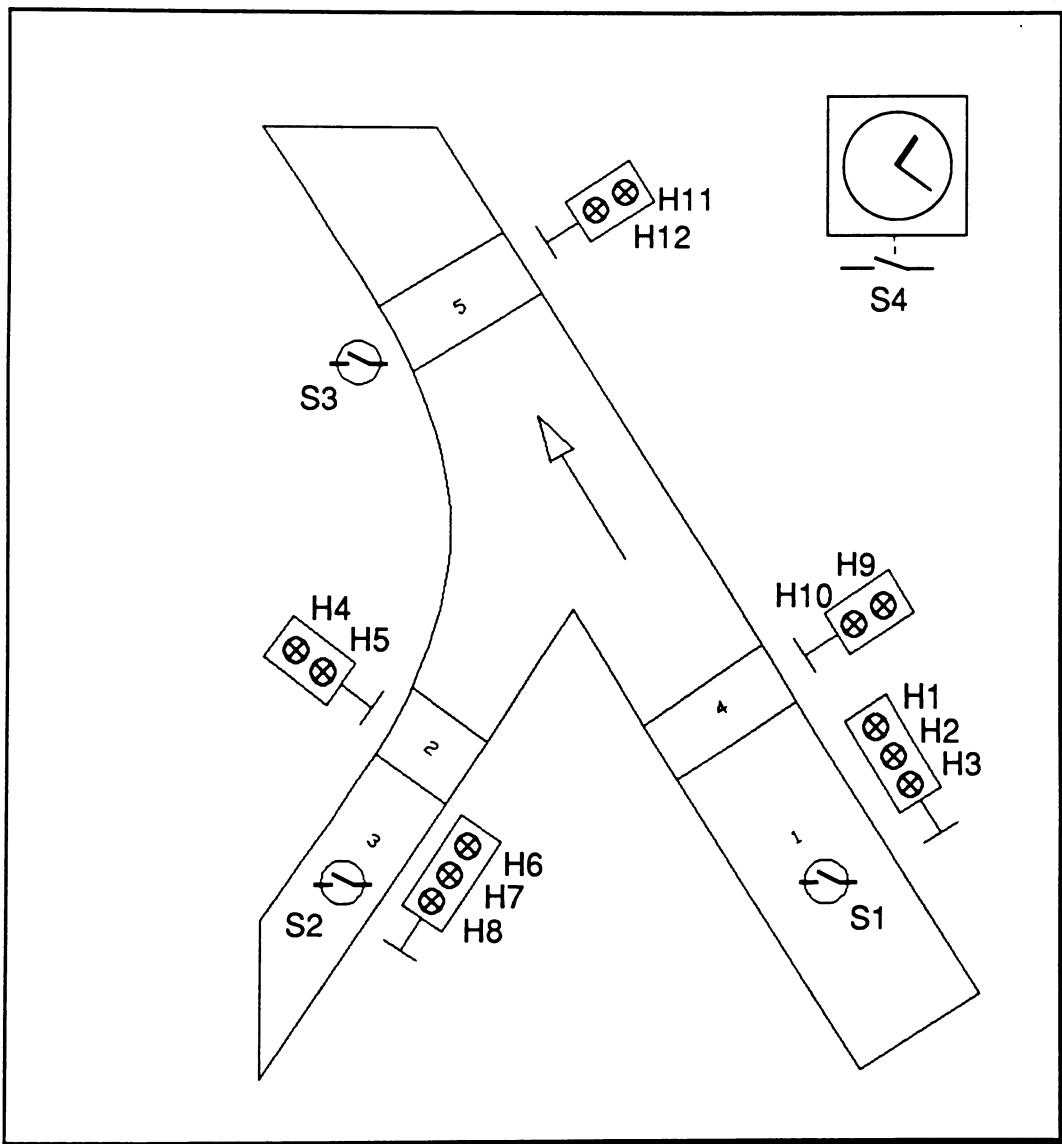


### 2.3 Das Technologieschema

Wie die einzelnen Betriebsmittel einer Anlage zusammenwirken, kann dem Technologieschema entnommen werden. Das Technologieschema gibt auch ungefähre Anhaltspunkte über die Lage der Betriebsmittel in der Anlage, ohne jeden Anspruch auf Maßstäblichkeit.

Das Technologieschema des Verkehrsampel-Modells (s. *Abbildung 2.2*) leitet sich aus einer kartographischen Aufsicht ab und entspricht somit weitgehend der Ansicht der Experimentierplatte (s. *Abbildung 2.3*).

Die Betriebsmittel werden in den folgenden Abschnitten erläutert.



*Abbildung 2.2:* Technologieschema der Ampelsteuerung

## 2. Beschreibung des Modells

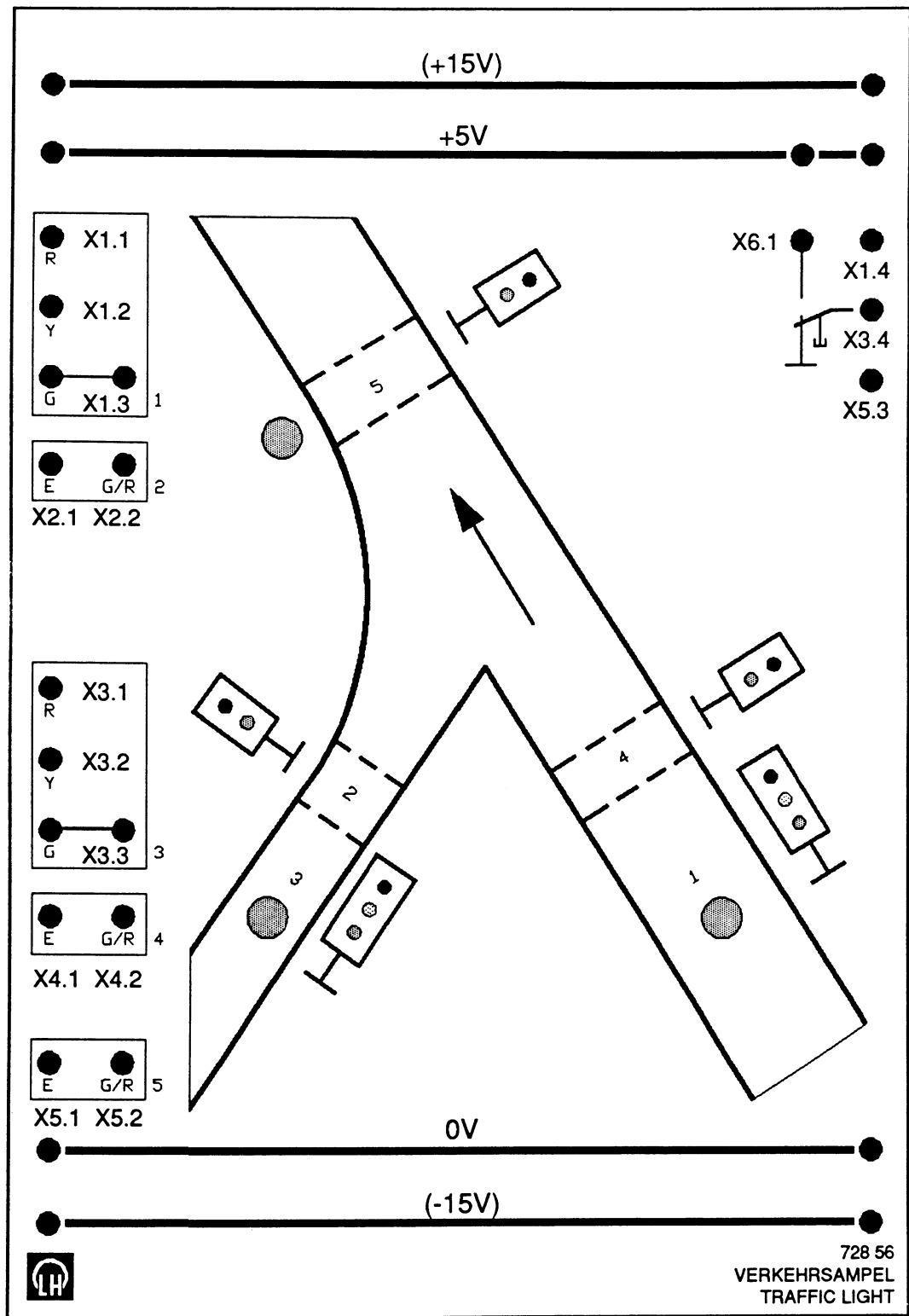


Abbildung 2.3: Ansicht des Experimentierplatten-Modells VERKEHRSAMPEL (Klemmenbezeichnungen hinzugefügt)

## **2.4 Die Verkehrsampeln**

Die Verkehrsampel 1 besitzt eine rote, eine gelbe und eine grüne Lampe. Die Lampen werden durch Signale an den dazugehörigen Buchsen gesteuert: Ein High-Signal läßt die Lampe leuchten, ein Low-Signal erlöschen. Eine Verriegelung unsinniger Schaltzustände (z.B. die Kombination gelb-grün) findet nicht statt. Diese Verriegelung muß der Steuerungscomputer in seinem Steuerungsprogramm gewährleisten.

Die Verkehrsampel 3 ist genauso wie die Verkehrsampel 1 aufgebaut. Die zu den Ampeln gehörigen Klemmenbezeichnungen und Bezeichnungen der Anschlußbuchsen sind der *Abbildung 2.3* und nachstehender Tabelle zu entnehmen.

Klemme	Bezeichnung Exp.platte	Funktion
X1.1 X1.2 X1.3	Feld 1, R Feld 1, Y Feld 1, G	Rote Lampe der Verkehrsampel 1 Gelbe Lampe der Verkehrsampel 1 Grüne Lampe der Verkehrsampel 1
X3.1 X3.2 X3.3	Feld 1, R Feld 1, Y Feld 1, G	Rote Lampe der Verkehrsampel 3 Gelbe Lampe der Verkehrsampel 3 Grüne Lampe der Verkehrsampel 3

---

## 2. Beschreibung des Modells

---

### 2.5 Die Fußgängerampeln

Die Fußgängerampel 2 besitzt eine rote und eine grüne Lampe. Sie wird durch Signale an zwei Buchsen gesteuert: Ein High-Signal an dem mit E bezeichneten Eingang X2.1 setzt die Fußgängerampel in Betrieb, ein Low-Signal schaltet sie ab.

**Hinweis:** Da bei dem Modell ein offener Eingang wie das Anlegen eines High-Signals wirkt, kann der Eingang unbeschaltet bleiben, wenn die Fußgängerampel immer benutzt werden soll.

Ein High-Signal an dem mit G/R bezeichneten Eingang X2.2 schaltet die grüne Lampe der Fußgängerampel ein und die rote Lampe aus. Umgekehrt schaltet ein Low-Signal am Eingang X2.2 die rote Lampe ein und die grüne Lampe aus.

Die Fußgängerampel besitzt somit eine Hardware-Verriegelung zur Vermeidung des unsinnigen Schaltzustands rot-grün.

In den nachfolgend gestellten Aufgaben (s. *Kapitel 4*) wird die Fußgängerampel 2 nicht eigens angesteuert. Durch einen Kupplungsstecker oder ein kurzes Laborkabel wird der Eingang X2.2 mit dem Eingang X1.3 verbunden (s. *Kapitel 3.3*). Damit wird die Fußgängerampel immer dann auf grün geschaltet, wenn die Verkehrsampel 1 ebenfalls grün zeigt. Eine solche Zwangskopplung nutzt zwar gegebenenfalls nicht alle Möglichkeiten der Freigabe des Überwegs aus, stellt aber (bei korrekter Programmierung der Verkehrsampeln 1 und 3) garantiert nur sichere Grün-Phasen her.

Die Fußgängerampel 4 entspricht in ihrem Aufbau und Anschluß der Fußgängerampel 2. Sie wird ebenfalls nur indirekt durch das Schaltsignal der grünen Lampe von Verkehrsampel 3 gesteuert.

Die Fußgängerampel 5 entspricht in ihrem Aufbau ebenfalls der Fußgängerampel 2. Der Eingang X5.2 wird jedoch durch eine eigene Signalleitung vom Steuerungsrechner gesteuert.

Klemme	Bezeichnung Exp.platte	Funktion
X2.1 X2.2	Feld 2, E Feld 2, G/R	Freigabe der Fußgängerampel 2 Grün / Rot-Steuerung Fußgängerampel 2
X4.1 X4.2	Feld 4, E Feld 4, G/R	Freigabe der Fußgängerampel 4 Grün / Rot-Steuerung Fußgängerampel 4
X5.1 X5.2	Feld 5, E Feld 5, G/R	Freigabe der Fußgängerampel 5 Grün / Rot-Steuerung Fußgängerampel 5

## **2.6 Die Melde-Einrichtungen**

Die "Induktionsschleifen" der Verkehrsampeln 1 und 3 liefern ein High-Signal, solange die Taste gedrückt wird, sonst ein Low-Signal.

Die Meldetaste der Fußgängerampel 5 liefert ebenfalls ein High-Signal, solange die Taste gedrückt wird, sonst ein Low-Signal.

Dies setzt voraus, daß der Eingang X6.1 durch einen Kupplungsstecker oder ein kurzes Laborkabel mit der Spannungsschiene +5V verbunden ist.

**Hinweis:** Die Tasten weisen keine Hardware-Einrichtung zur Elimination von Prellerscheinungen auf. Soll die Anzahl der Tastenbetätigungen mit einem Maschinenprogramm ermittelt werden (was in den Aufgabenstellung des Kapitels 4 nicht vorkommt), so müßte das Programm die Prellerscheinungen abfangen.

Klemme	Bezeichnung Exp.platte	Funktion
X1.4	rechts, 1	Tasteranschluß "Induktionsschleife 1"
X3.4	rechts, 3	Tasteranschluß "Induktionsschleife 3"
X5.3	rechts, 5	Tasteranschluß "Meldetaste 5"

## **2.7 Die Schaltuhr**

Die Schaltuhr dient der Steuerung des Tag/Nacht-Betriebs. Sie ist im Modell nicht ausgeführt, sondern wird durch den Schalter am Eingang E03 der 8-Bit-Parallel-Eingabe simuliert.

Es wird angenommen, daß die Schaltuhr zur Tageszeit ein Low-Signal und zur Nachtzeit ein High-Signal liefert.

---

## 2. Beschreibung des Modells

---

### 2.8 Die Funktionsbeschreibung

Die Funktionsbeschreibung den vom Planer der Anlage beabsichtigten Ablauf oder die Funktion deutlich. Ohne die Funktionsbeschreibung wird es kaum gelingen, ein Steuerungsprogramm für die Anlage zu entwickeln oder eine gestörte Anlage wieder in Betrieb zu setzen.

Die Funktionsbeschreibung ist im vorliegenden Fall insbesondere das Pflichtenheft, das mit der Lösung der in Kapitel 4 beschriebenen Aufgaben Stück für Stück erfüllt wird. Daher sollte bei der Bearbeitung der Aufgaben immer wieder die Funktionsbeschreibung zu Rate gezogen werden, um sich nicht in technischen Details zu verrennen, sondern den Gesamtblick für die Anlage zu wahren.

Funktionsbeschreibung:

Die Verkehrsampeln 1 und 3 sowie die Fußgängerampel 5 sollen so gesteuert werden, daß entsprechend dem jeweiligen Verkehrsaufkommen ein konfliktfreier und optimaler Verkehrsfluß im Einmündungsbereich gewährleistet ist.

Die Fußgängerampeln 2 und 4 werden jeweils mit den Grünphasen der Verkehrsampeln 1 und 3 gekoppelt, also nicht eigens angesteuert. Sie werden lediglich für den Betrieb freigegeben oder gesperrt. In diesem speziellen Fall werden somit bewußt Einschränkungen in Kauf genommen.

Nachfolgende Schaltungsbedingungen gelten während der Tageszeit für jede der Ampeln 1, 3 oder 5:

1. Die Freigabe eines Verkehrswegs beginnt mit der Rot-Gelb-Phase. Die Bedingungen, unter denen der Verkehrsweg freigegeben wird, sind nachstehend erläutert. Die Rot-Gelb-Phase dauert eine festgesetzte Zeit an. Bei der Fußgängerampel 5 wird die Rot-Gelb-Phase sinngemäß durch eine Übergangsphase ersetzt, in der nur die rote Lampe leuchtet.
2. Es folgt die Grün-Phase. Die Grün-Phase bleibt zunächst für eine Mindestdauer bestehen. Die Mindestdauer wird entsprechend des Verkehrsaufkommens festgelegt, wobei jeweils zwei Zeiten zur Verfügung stehen, von denen eine doppelt so lang wie die andere ist. Die lange Zeit wird bei hoher Verkehrsdichte auf diesem Weg, die kurze bei normaler Verkehrsdichte gewählt.
3. Nach Ablauf der Mindestdauer der Grün-Phase wird geprüft, ob von einem anderen Verkehrsweg Verkehrsaufkommen gemeldet wird. Melden beide anderen Verkehrswege kein Verkehrsaufkommen, wird die Grün-Phase mit fortgesetzter Prüfung verlängert.

4. Wenn Verkehrsaufkommen von einem anderen Verkehrsweg gemeldet wurde, wird die Sperrung eingeleitet. Es folgt zunächst die Gelb-Phase, die dazu dient, den Einmündungsbereich zu räumen. Die Gelb-Phase dauert eine festgesetzte Zeit an. Bei der Fußgängerampel ist die Gelb-Phase durch eine Übergangsphase ersetzt, in der nur die rote Lampe leuchtet.
5. Anschließend kann ein anderer Verkehrsweg freigegeben werden. Liegen Meldungen beider anderen Verkehrswege vor (dies ist der Fall, wenn die Meldungen praktisch gleichzeitig eingingen oder aber bereits während der Mindestdauer der Grün-Phase vorlagen und gespeichert wurden), so werden die Verkehrswege in folgender Reihenfolge bedient: 1 - 3 - 5.
6. Für die betrachtete Ampel beginnt nun die Rot-Phase. Deren Dauer hängt davon ab, welche anderen Verkehrswege nun die Freigabe erhalten, und ob an der betrachteten Ampel Fahrzeuge bzw. Fußgänger eintreffen. Um letzteres festzustellen, wird permanent die Melde-Einrichtung überprüft. Trifft sofort nach Beginn der Rot-Phase eine Meldung ein (z.B. innerhalb von 5 Sekunden), so wird diese gespeichert und damit hohe Verkehrsdichte registriert. Trifft die Meldung zu einem späteren Zeitpunkt ein, so liegt normale Verkehrsdichte vor. Die Meldung muß eventuell gespeichert werden, wenn ein anderer Verkehrsweg noch vorher bedient werden muß. Sind die anderen Verkehrswege jedoch bereits entsprechend der Mindestdauer der Grün-Phasen freigeschaltet, so kann der Anforderung sofort stattgegeben werden (vergl. Punkte 3 und 4).

Während der Nacht sind die Fußgängerampeln gänzlich abgeschaltet. Die Verkehrsampeln weisen ein gelbes Blinklicht auf, um die Verkehrsteilnehmer auf die besondere Situation aufmerksam zu machen.

Folgende Schaltzeiten sind eingestellt:

Grün-Phase der Verkehrsampel 1: mind. 15 bzw. 30 Sekunden,

Grün-Phase der Verkehrsampel 3: mind. 10 bzw. 20 Sekunden,

Grün-Phase der Fußgängerampel 5: mind. 10 bzw. 20 Sekunden,

Gelb-Phasen aller Ampeln: 3 Sekunden,

Rot-Gelb-Phasen aller Ampeln: 3 Sekunden.

Die Rot-Phasen der Ampeln ergibt sich aus den übrigen Phasen.

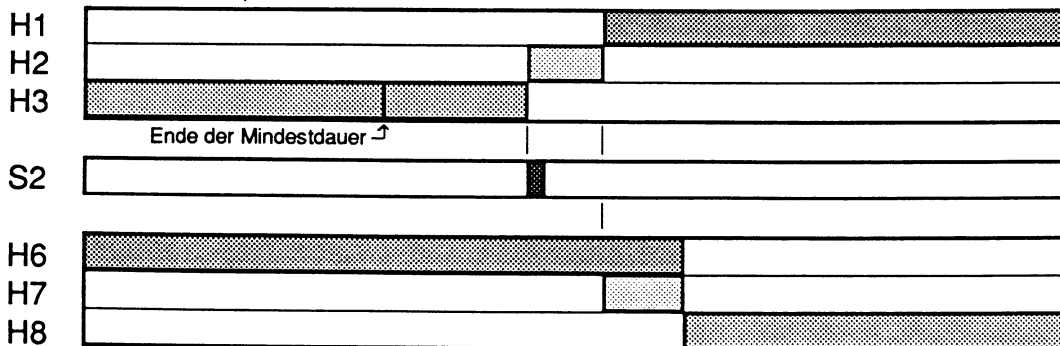
Kriterium für hohe Verkehrsdichte: Meldung innerhalb von 5 Sekunden.

**Hinweis:** Die Grün-Phasen sind eher kürzer als in Realität festgelegt, um ohne allzu große Wartezeiten die Steuerungsprogramme testen zu können.

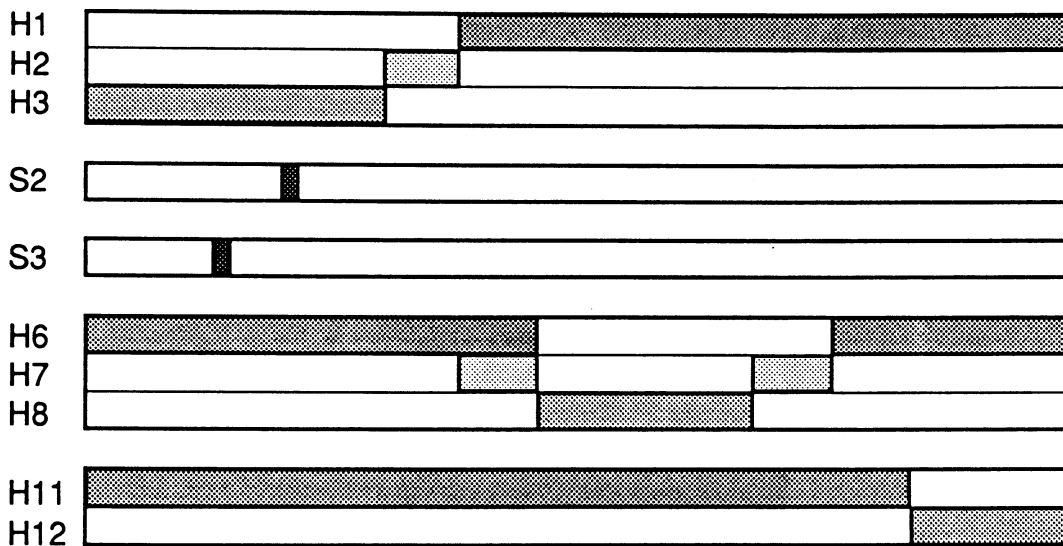


## 2. Beschreibung des Modells

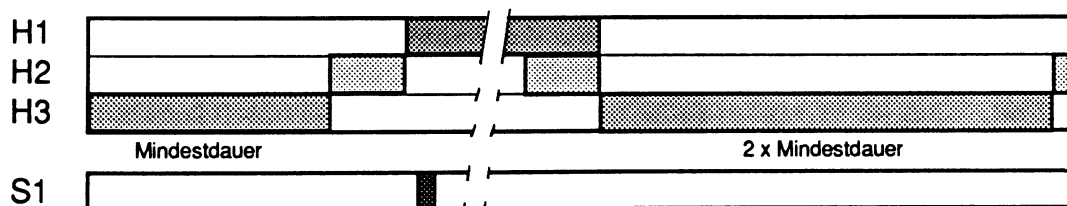
Zur Illustration obiger Vorschriften sind die Signal-Zeit-Diagramme für drei ausgesuchte Fälle im Zusammenspiel der Ampeln dargestellt:



Die Grün-Phase von Ampel 1 dauert an, bis eine Meldung von anderen Verkehrswegen vorliegt.



Die Grün-Phase endet nach Ablauf der Mindestdauer, weil bereits andere Meldungen vorliegen; entsprechend der Festlegung wird zuerst Ampel 3 und dann Ampel 5 bedient (obwohl die Meldungen umgekehrt eingingen).



Sofort nach Ende der Gelb-Phase liegt bereits wieder eine Meldung vor; die nächste Mindestdauer der Grün-Phase wird verdoppelt.

---

### 3. Der Anschluß an den MFA-Mikrocomputer

#### 3.1 Liste der erforderlichen Baugruppen

Zur Steuerung des Verkehrsampel-Modells (Artikel Nr. 728 56 der LEY-BOLD DIDACTIC GmbH) wird der MFA-Mikrocomputer (vertrieben von der vgs Verlagsgesellschaft) eingesetzt. Der MFA-Mikrocomputer besteht aus einem modularen System von Baugruppen und kann für die jeweilige Anwendung "maßgeschneidert" werden. Welche Ausstattung an Baugruppen Sie benötigen, hängt u.a. von der benutzten Programmiersprache ab.

Bei der Programmierung in Assembler (Maschinensprache) sind mindestens folgende Baugruppen des MFA-Mikrocomputers vonnöten:

- Baugruppenträger mit Bus-Verdrahtung, Bus-Abschluß, Trafo-Einschub und Spannungsregelung,
- CPU-Baugruppe (Prozessor 8085),
- Betriebsprogramm MAT 85 auf einer 16-K-Karte,
- 8-K-RAM-Karte (ab Adresse E000) mit 8 K RAM,
- Video-Interface, ASCII-Tastatur und Monitor mit Cinch-Anschluß.

Zum Anschluß des Modells:

- 8-Bit-Parallel-Eingabe,
- 8-Bit-Parallel-Ausgabe.

Diese beiden Baugruppen arbeiten auf der Basis von TTL-Signalen wie das Verkehrsampel-Modell auch. Die MFA-Ausgabe-Baugruppe mit galvanisch getrennten Ausgängen kann hier nicht verwendet werden, da sie nur Spannungen im Bereich von 7 bis 24 V schalten kann. Soll die galvanisch getrennte Eingabebaugruppe anstelle der hier empfohlenen Baugruppe mit TTL-Eingängen eingesetzt werden, so müssen die Kabelverbindungen zwischen den Laborbuchsen des Modells und den Klemmen der Baugruppe selbst hergestellt werden (Masseanschluß nicht vergessen!).

Bei der Arbeit mit dem Assembler des MFA-Mikrocomputers ist die Verwendung des BFZ-Editors und der Floppy-Disk sehr zu empfehlen. In den Aufgabenstellungen dieses Handbuchs wird das Steuerungsprogramm Schritt für Schritt entwickelt. Es wird dabei immer nur die jeweils erzielte Verbesserung diskutiert, die in das aus der vorigen Aufgabenlösung bestehende Programm eingebaut wird. Eine solche Vorgehensweise weist große Praxisnähe auf, erfordert aber auch den Umgang mit einem Texteditor, insbesondere wegen der nachträglichen Einfügungen von Programmzeilen. Dies wäre mit den Möglichkeiten des Betriebsprogramms MAT-85 nur schwer zu bewerkstelligen. Es wird daher folgende Zusatzausstattung empfohlen:

- Software-Erweiterung SP 1 (auf der MAT-85-Karte einzusetzen),
- Floppy-Disk-Interface,
- BFZ-Mini-DOS (4 K EPROM, mit dem Floppy-Interface mitgeliefert, einzusetzen bei Adresse 4000 bis 4FFF, vorteilhaft auf einer 16-K-Karte),

### 3. Der Anschluß an den MFA-Mikrocomputer

- Diskettenstation (Laufwerk mit Gehäuse, Netzteil und Anschlußkabel),
- BFZ-Editor; einzusetzen ab Adresse 5000 auf derselben 16-K-Karte wie das Mini-DOS.
- Mindestens 8k RAM ab Adresse 6000 als Arbeitsspeicher für den Texteditor.

Für die Lösung der Aufgaben durch SPS-Programme wird die oben erwähnte Software-Erweiterung SP 1 benötigt. Zur Programmdiagnose kann sehr wirkungsvoll die Zeitwerk-Baugruppe eingesetzt werden:

- Zeitwerk (4fach).

Für die Lösung der Aufgaben in BFZ-Steuer-BASIC ist ebenfalls die oben erwähnte Software-Erweiterung SP 1 nötig. Zusätzlich ist erforderlich:

- 16-K-RAM-Karte (Basis-Adresse 4000) mit mindestens 4 K RAM ab Adresse 6000; hierfür ist dieselbe 16-K-Karte zu benutzen, die das Mini-DOS trägt.

Adresse		Adresse		Adresse	
0000	Betriebsprogramm MAT 85	0000	Betriebsprogramm MAT-85	0000	Zero-Page
1FFF		1FFF		00FF	
2000	Softwarepaket SP1 (SPS, BASIC)	2000	Softwarepaket SP1 (SPS, BASIC)	0100	transient program area TPA
3FFF		3FFF			
4000	BFZ-Mini-DOS	4000	BFZ-Mini-DOS		
4FFF		4FFF		DBFF	
5000	BFZ-Editor	5000	Kontaktplanausgabe	DC00	console command processor CCP
5FFF		57FF			
6000	Minimalspeicher für BASIC	5800	SPS-Dokumentation	E3FF	
67FF		5FFF		E400	basic disk operating system BDOS
6800	zusätzlicher Speicher für BASIC	6000	reserviert		
DFFF		67FF			
E000	Speicher für SPS und BFZ-MINI-DOS	6800	BFZ-Editor		
F7FF		77FF		F1FF	
F800	Mindestspeicherausbau für MAT 85	7800	Editor Hilfsmenü	F200	basic input output system BIOS
FFFF		7FFF			
		8000	32 K RAM für alle Softwarepakete		
		FFFF		FFFF	

Abbildung 3.1: Speicherausbau des MAT 85-Systems (links), des MAT 32 K-Systems (Mitte) und des CPM-Systems (rechts).

Sehr flexibel ist die Verwendung der 64-K-Speicherbaugruppe mit dem MAT32K sowie 32 K RAM. Das MAT32K-EEPROM enthält alle genannten Softwarepakete (MAT 85, SP 1, Mini-DOS, Editor) sowie zusätzlich die Kontaktplanausgabe und Dokumentationsfunktion (Editiermöglichkeit) zum SPS-Interpreter.

Generell wird für die Programmierung unter allen drei Programmiermethoden (Assembler, SPS und BASIC) die Verwendung eines Diskettenlaufwerks (s.o.) dringend empfohlen. Zu den Vorteilen der hohen Datensicherheit, des schnellen Speicherns und Ladens kommt hinzu, daß dann die Beispielpprogramme der diesem Handbuch beigelegten Diskette jederzeit zur Verfügung stehen.

Der MFA-Mikrocomputer kann (nach einem geringfügigen Umbau) auch unter dem Betriebssystem CP/M betrieben werden. Die notwendige Geräteausrüstung entnehmen Sie der CP/M-Anleitung des MFA-Medien-systems.

Die diesem Handbuch beigelegte CP/M-Diskette umfaßt Programme, die mit dem CP/M-Assembler ASM und CP/M-Ladeprogramm LOAD bzw. dem MBASIC-Interpreter erstellt wurden. ASM und LOAD sind Bestandteil des MFA-CP/M-Pakets. Dagegen ist MBASIC im Microsoft-Entwicklungspaket enthalten, das (wie das gesamte MFA-Medien-system) ebenfalls von der vgs Verlagsgesellschaft zu beziehen ist.

Die mit ASM/LOAD bzw. MBASIC erstellten Programme werden im folgenden nicht ausdrücklich besprochen, denn sie unterscheiden sich nur unwesentlich von den MAT 85- bzw. den Steuer-BASIC-Programmen. Zur Benutzung der CP/M-Programme sollte Erfahrung im Umgang mit CP/M vorhanden sein.

Als Grundlage des hier vorliegenden Kurses ist die Dokumentation des MFA-Mikrocomputersystems unbedingt notwendig. Insbesondere sollten die Fachpraktischen Übungen Band 1 (FPÜ1) und die Fachtheoretischen Übungen (FTÜ) stets zur Hand sein.

---

### 3. Der Anschluß an den MFA-Mikrocomputer

---

#### 3.2 Modifikation des MFA-Mikrocomputers

Wenn die Aufgaben des Kapitels 4 mit Hilfe des BFZ-SPS-Interpreters gelöst werden sollen, so ist nachstehende Modifikation des MFA-Mikrocomputers durchzuführen, um die Software-Timer benutzen zu können.

##### **Modifikation der Bus-Platine:**

Das Rechtecksignal, das am Anschluß 23 der Spannungsregelung zur Verfügung steht, wird auf den RST-7.5-Eingang des Mikroprozessors durchgeschaltet. Dazu wird auf der Bus-Platine der Anschluß, der mit dem Impulssymbol bezeichnet ist, mittels eines dünnen isolierten Schaltahts mit der Busleitung 25a verbunden.

Die genaue Lage der Anschlüsse ist in *FTÜ, Kapitel 7.2, S. 144* aufgezeichnet.

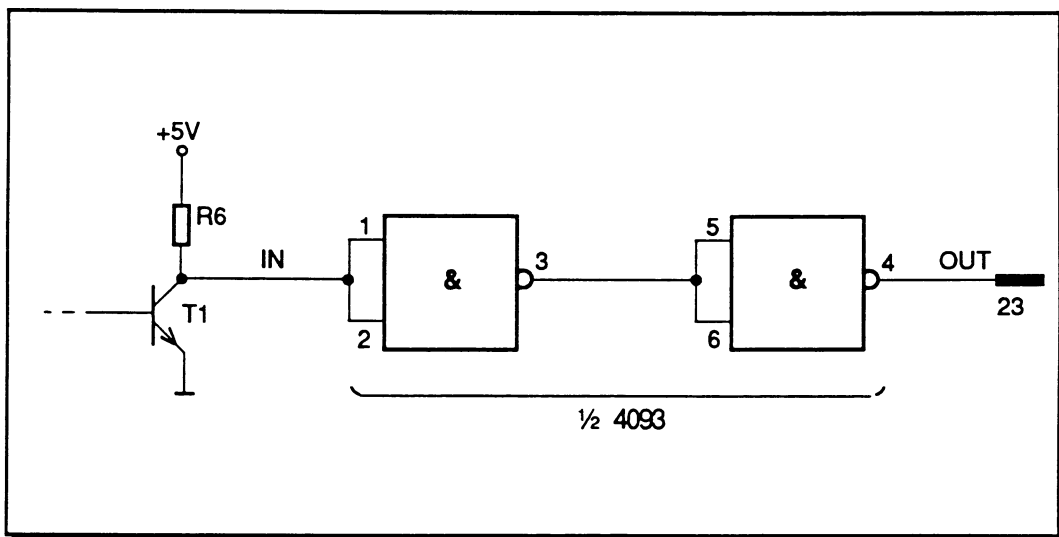
Will man im Rahmen anderer Anwendungen des MFA-Mikrocomputers den RST-7.5-Eingang auch aus anderen Signalquellen (z.B. dem Zähler/Zeitgeber-Einschub) bedienen, so empfiehlt es sich, diese Verbindung steckbar oder schaltbar auszuführen.

### Modifikation des Spannungsregelungs-Einschubs:

Das Rechtecksignal des Spannungsregelungs-Einschubs muß mit Hilfe eines CMOS-Schmitt-Triggers geformt werden, um ein mehrfaches Ansprechen des RST-7.5-Eingangs des Mikroprozessors zu vermeiden. Dies führt dazu, daß die Software-Timer zu schnell laufen und deshalb zu kurze Zeitintervalle erzeugen.

Das CMOS-IC 4093 (4-fach NAND mit Schmitt-Trigger-Eingängen) wird mit seiner Oberseite auf die Bestückungsseite der Leiterplatte des Spannungsregelungs-Einschubs geklebt. Die Leitung zwischen dem Kollektor des Transistors T1 und dem Anschluß 23 wird unterbrochen. Eine zweistufige Triggerschaltung auf der Basis des 4093 wird an dieser Stelle laut Schaltbild in *Abbildung 3.2* eingefügt.

Die genaue Lage der Anschlüsse ist in *FTÜ, Kapitel 7.2, S. 145f* gezeigt.



*Abbildung 3.2:* Pulsformung des Rechtecksignals mittels zweier Triggerstufen.

---

## 3. Der Anschluß an den MFA-Mikrocomputer

---

### 3.3 Die Zuordnungsliste

In einer Steuerung werden verschiedene Betriebsmittel eingesetzt. Diese werden laut Norm durch Kennbuchstaben bezeichnet. Im vorliegenden Modell werden Schalter und Taster verwendet, die mit dem Kennbuchstaben S bezeichnet sind. Die Anzeigeelemente werden mit H bezeichnet.

Die Betriebsmittel sind über Anschlüsse zugänglich, von denen Leitungen zum Steuerungscomputer führen. Die Anschlüsse werden laut Norm mit dem Kennbuchstaben X bezeichnet. Im Falle industrieller Anlagen werden die Anschlüsse als Klemmenreihen ausgeführt, im vorliegenden Modell sind wegen des flexiblen Auf- und Abbaus jedoch Laborbuchsen eingesetzt.

Die Anschlüsse am Steuerungscomputer werden mit den Kennbuchstaben A für die Ausgänge und E für die Eingänge bezeichnet (diese Bezeichnung ist nicht genormt, manchmal werden dafür auch Q und I oder Y und X verwendet). Der Anschlußnummer folgt eine zweistellige Zahl, deren erste Ziffer die Port-Adresse der Eingabe- oder Ausgabebaugruppe darstellt (in dem vorliegenden Fall immer 0) und deren zweite Ziffer die Bitposition im Datenwort bezeichnet (0 bis 7).

**Beispiel:** A02 bezeichnet den Ausgang, der mit Bit 2, Adresse 0 angesteuert wird. Dieser Ausgang befindet sich an der dritten Buchse von oben auf der Frontplatte der 8-Bit-Parallel-Ausgabe des MFA-Systems.

Der Zusammenhang zwischen Betriebsmittel, Anschlußbuchse des Betriebsmittels und Anschlußbuchse des Steuerungscomputers wird durch die Zuordnungsliste (s. *Abbildung 3.3*) hergestellt. In dieser findet sich auch noch ein stichwortartiger Hinweis auf die Funktion des jeweiligen Betriebsmittels. Die Zuordnungsliste ist (zusammen mit der Funktionsbeschreibung, dem Technologieschema und dem Steuerungsprogramm) unentbehrlicher Teil der Dokumentation jeder Steuerungsanlage. Sie sollte auch bei der Erstellung des Steuerungsprogramms immer griffbereit sein.

**Hinweis:** Die Zeitschaltuhr S4 (an Eingang E03) ist im Modell nicht vorhanden. Anstelle des Schalters der Uhr wird der Schalter auf der Frontplatte der 8-Bit-Parallel-Eingabe benutzt.



Kenn- zeichen	Ein-/ Ausgang (Computer)	Klemme (Modell)	Betriebs- mittel	Funktion
S1	E00	X1.4	Indukt.- schleife	Ankunft eines Fahrzeugs an Ampel 1
S2	E01	X3.4	Indukt.- schleife	Ankunft eines Fahrzeugs an Ampel 3
S3	E02	X5.3	Taster	Anforderung Überweg 5
S4	E03	--	Schalter	Zeitschaltuhr Tag / Nacht
H1	A00	X1.1	Lampe	Anzeige "rot" Ampel 1
H2	A01	X1.2	Lampe	Anzeige "gelb" Ampel 1
H3	A02	X1.3	Lampe	Anzeige "grün" Ampel 1
H4 / H5	A07	X2.1	Signal	Betriebsfreigabe Ampel 2
"	A02	X2.2		"rot"/"grün" Ampel 2
H6	A03	X3.1	Lampe	Anzeige "rot" Ampel 3
H7	A04	X3.2	Lampe	Anzeige "gelb" Ampel 3
H8	A05	X3.3	Lampe	Anzeige "grün" Ampel 3
H9 / H10	A07	X4.1	Signal	Betriebsfreigabe Ampel 4
"	A05	X4.2		"rot"/"grün" Ampel 4
H11/ H12	A07	X5.1	Signal	Betriebsfreigabe Ampel 5
"	A06	X5.2		"rot"/"grün" Ampel 5

Abbildung 3.3: Zuordnungsliste zur Steuerung des Verkehrsampel-Modells mit dem MFA-Mikrocomputer

### 3. Der Anschluß an den MFA-Mikrocomputer

#### 3.4 Anschluß an die E/A-Baugruppen

Der Anschluß des Verkehrsampel-Modells erfolgt an die Baugruppen 8-Bit-Parallel-Ausgabe und 8-Bit-Parallel-Eingabe. Beide Baugruppen müssen vor dem Einsetzen in den MFA-Mikrocomputer auf Port-Adresse 0 eingestellt werden (s. *FPÜ 1, Kapitel 4.1 und 4.2*).

Wenn Sie noch keine Erfahrung mit den Baugruppen gesammelt haben, sollten Sie die Funktionsprüfung nach *FPÜ1, Kapitel 4.1 und 4.2* vornehmen. Überzeugen Sie sich auch mit Hilfe des IN- und OUT-Kommandos des MAT-85-Betriebssystems von der Funktionstüchtigkeit der beiden Baugruppen (Kommandobeschreibung siehe *FTÜ, Kapitel 7.1, Seite 58f*). Der Anschluß des Modells erfolgt mit Laborkabeln mit 4-mm-Steckern laut Zuordnungsliste. Zur Herstellung des Bezugspotentials muß weiterhin eine Masseverbindung zwischen der 0-V-Schiene des Verkehrsampel-Modells und der 0-V-Buchse des Spannungsregelungs-Einschubs des MFA-Mikrocomputers hergestellt werden.

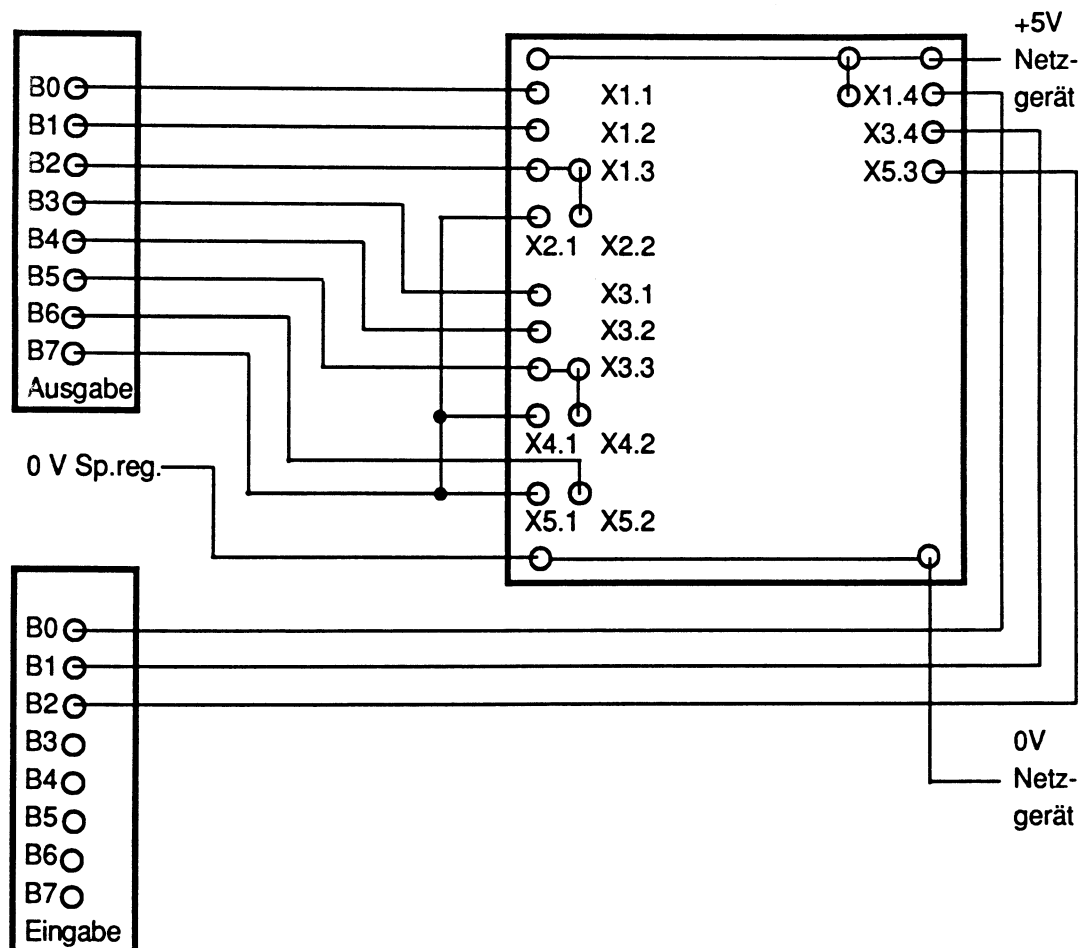


Abbildung 3.4: Schaltbild zum Anschluß an die E/A-Baugruppen

### 3.5 Stromversorgung des Verkehrsampel-Modells

Das Verkehrsampel-Modell wird mit 5 V Gleichspannung versorgt. Die Stromversorgung muß bis zu 0,15 A belastbar sein, ohne daß die Qualität der Siebung der Spannung leidet oder die Spannung 4,75 V unterschreitet. Da Überspannung zu einer Zerstörung der Elektronik des Verkehrsampel-Modells führt, wird empfohlen, kein einstellbares Netzgerät, sondern ein Netzgerät mit Festspannungsausgang, z.B. LEYBOLD DIDACTIC 726 84, zu verwenden.

Bei Wahrung der nötigen Vorsichtsmaßnahmen kann die Versorgungsspannung auch dem Spannungsregelungs-Einschub des MFA-Computers entnommen werden.

---

### 3. Der Anschluß an den MFA-Mikrocomputer

---

#### 3.6 Programme auf der Diskette im BFZ-Mini-DOS-Format

Die beigelegte Diskette enthält alle Musterlösungen der Aufgaben jeweils als Assembler-Quelltext (ladbar mit dem BFZ-Editor), als ausführbares Maschinenprogramm, als BFZ-Steuer-BASIC-Programm für zwei verschiedene Speicherbelegungen und als SPS-Programm, letzteres in der Variante, wie sie in den FTÜ beschrieben ist, sowie in der Variante nach DIN und als kommentierte Textdatei. Der erste Namensteil des Dateinamens kennzeichnet die Aufgabenstellung. Der zweite Namensteil (die drei Zeichen nach dem Punkt) bezeichnet die Dateiarart. Dabei gilt:

<b>.ASM</b>	Assembler-Quelltext für den BFZ-Editor,
<b>.MAT</b>	unter MAT-85 ausführbares Maschinenprogramm,
<b>.BAS</b>	BASIC-Datei für den BFZ-Steuer-BASIC-Interpreter des MAT 32 K,
<b>.B6K</b>	BASIC-Datei für den BFZ-Steuer-BASIC-Interpreter des MAT 85,
<b>.DOK</b>	Textdatei mit kommentiertem SPS-Programm,
<b>.SPS</b>	SPS-Datei für den BFZ-SPS-Interpreter,
<b>.DIN</b>	SPS-Datei für den BFZ-SPS-Interpreter nach DIN.

Die beiden Varianten der BASIC-Programme enthalten identischen Programmtext. Da BFZ-Steuer-BASIC-Interpreter des MAT 32 K eine Speicherung ab Adresse 8000 (hex) fordert, jener des MAT 85 jedoch ab Adresse 6000 (hex) und eine Verschiebung im Speicher mühsam und fehlerträchtig ist, werden beide Varianten auf der Diskette zur Verfügung gestellt. Zur Nutzung des Arbeitsspeichers siehe *Abbildung 3.1, Seite 18*.

#### Beispiele:

Die Datei A5C.ASM ist der Assembler-Quelltext der Lösung der Aufgabe A5c.

Die Datei A5C.BAS ist das BASIC-Programm zur Lösung der gleichen Aufgabe A5c.

Die Datei A10B.DIN ist das SPS-Programm (nach DIN-Norm) zur Lösung der Aufgabe A10b.

Die Diskette enthält folgende Dateien:

Die Hilfestellung zu Aufgabe A4  
(Anzeige der Eingangsleitungen):

**A4.ASM**

**A4.MAT**

Die Beispielprogramme zu Aufgabe A5  
(Steuerung der Verkehrsampeln):

**A5C.ASM**

**A5C.MAT**

**A5D.ASM**

**A5D.MAT**

**A5C.BAS**

**A5C.B6K**

**A5D.BAS**

**A5D.B6K**

**A5C.DOK**

**A5C.DIN**

**A5C.SPS**

Die Beispielprogramme zu Aufgabe A6  
(Steuerung der Verkehrs- und Fußgängerampeln):

**A6C.ASM**

**A6C.MAT**

**A6D.ASM**

**A6D.MAT**

**A6C.BAS**

**A6C.B6K**

**A6D.BAS**

**A6D.B6K**

**A6D.DOK**

**A6D.SPS**

**A6D.DIN**

Die Lösung der Aufgabe A7  
(Bedarfssteuerung der Nebenstraße):

**A7B.ASM**

**A7B.MAT**

**A7B.BAS**

**A7B.B6K**

**A7B.DOK**

**A7B.DIN**

**A7B.SPS**

---

### **3. Der Anschluß an den MFA-Mikrocomputer**

---

Das Beispielprogramm zu Aufgabe A8  
(Bedarfssteuerung der Nebenstraße und des Fußwegs):

**A8D.ASM**  
**A8D.MAT**  
**A8D.BAS**  
**A8D.B6K**  
**A8D.DOK**  
**A8D.DIN**  
**A8D.SPS**

Das Beispielprogramm zu Aufgabe A9  
(Speicherung der Meldeimpulse):

**A9B.ASM**  
**A9B.MAT**  
**A9B.BAS**  
**A9B.B6K**  
**A9B.DOK**  
**A9B.DIN**  
**A9B.SPS**

Das Beispielprogramm der Aufgabe A10  
(Verkehrsdichten-abhängige Steuerung):

**A10B1.ASM + A10B2.ASM**  
**A10B.MAT**  
**A10B.BAS**  
**A10B.B6K**  
**A10B.DOK**  
**A10B.DIN**  
**A10B.SPS**

Das Beispielprogramm der Aufgabe A11  
(Tag/Nacht-Schaltung) sowie der Ausgangsprogrammtext (s. *Kapitel 4.11*)

<b>A11B1.ASM + A11B2.ASM</b>	<b>A11B1.ASM + A11BLEER.ASM</b>
<b>A11B.MAT</b>	<b>A11BLEER.MAT</b>
<b>A11B.BAS</b>	<b>A11BLEER.BAS</b>
<b>A11B.B6k</b>	<b>A11BLEER.B6K</b>
<b>A11B.DOK</b>	<b>A11BLEER.DOK</b>
<b>A11B.DIN</b>	<b>A11BLEER.DIN</b>
<b>A11B.SPS</b>	<b>A11BLEER.SPS</b>

### 3.7 Programme auf der Diskette im BFZ-CP/M-Format

Alle Aufgabenlösungen in Assembler und BASIC liegen auch als CP/M-Dateien vor. Die Assemblerprogramme wurden mit der CP/M-Version des BFZ-Editors BFZED erstellt, mit dem 8080-Assembler ASM assembliert und dem Programm LOAD in eine ausführbare Datei umgewandelt. Die Programme BFZED, ASM und LOAD sind Bestandteil der BFZ-CP/M-Systemdiskette.

Die Unterschiede der CP/M-Assemblerprogramme zu den Programmen für den BFZ-Assembler sind geringfügig. Wichtigster Unterschied ist die Verschiebung des Programmanfangs auf den Beginn des TPA (transient program area - Startadresse 0100 hex.). Die Zahlenausgabe in **A4.COM** wird durch CP/M-BDOS-Funktionen bewerkstelligt.

Die BASIC-Programme wurden mit dem Microsoft-BASIC-Interpreter MBASIC geschrieben, der Bestandteil des Microsoft-Entwicklungspakets ist, das bei der vgs Verlagsgesellschaft getrennt erworben werden muß. Eigenschaften des MBASIC, die über das BFZ-Steuer-BASIC hinausgehen, wurden bewußt nicht eingesetzt, um auch hier wieder die Programme vergleichbar zu halten. Die Zahlenkonstanten der FOR..NEXT-Schleifen zur Bestimmung des 1/10-Sekunden-Takts fallen bei MBASIC wegen der unterschiedlichen Arbeitsweise des Interpreters anders aus. So wird eine leere FOR..NEXT-Schleife von MBASIC langsamer abgearbeitet. Sobald jedoch verschiedene Eingabe- und Rechenoperationen in der Schleife enthalten sind, wird diese von MBASIC schneller als unter BFZ-Steuer-BASIC ausgeführt.

Die Programme sind wegen der geringen Unterschiede in dieser Dokumentation nicht abgedruckt. Im Bedarfsfall können sie auf einem Drucker ausgegeben werden.

Die Bezeichnung der Dateien im ersten Namensteil ist identisch mit jener der BFZ-Mini-DOS-Diskette. Die Bezeichnungen des zweiten Namensteils kennzeichnen die Art der Datei:

- |             |  |
|-------------|--|
| <b>.ASM</b> | Quelltexte für den 8080-Assembler ASM (erstellt mit dem Texteditor BFZED). |
| <b>.COM</b> | unter CP/M ausführbare Maschinenprogramme.                                 |
| <b>.BAS</b> | unter MBASIC ausführbare BASIC-Programme.                                  |



**4.1 Aufgabe A1: Aufbau des Modells**

Für diese und die folgenden Aufgaben wird der MFA-Mikrocomputer in der in *Kapitel 3.1* beschriebenen Ausbaustufe sowie das Modell Verkehrsampel benötigt. Die Kabelverbindungen werden laut Zuordnungsliste (*Kapitel 3.2*) und Schaltbild (*Kapitel 3.3*) hergestellt. Dazu werden zwölf lange Laborkabel, die vom MFA-Mikrocomputer zum Modell führen, und fünf kurze Laborkabel (davon können drei Brückenstecker sein) für Verbindungen innerhalb des Modells benötigt. Weiterhin werden zwei Brückenstecker zum Anschluß des stabilisierten Netzgeräts LEYBOLD-DIDACTIC 726 84 benötigt oder ein langes Laborkabel, wenn der Spannungsregelungs-Einschub des MFA-Mikrocomputers benutzt wird.

- Die Massebuchse des Spannungsregelungs-Einschubs wird mit der 0-V-Schiene des Verkehrsampel-Modells verbunden.
- Die Eingänge des Verkehrsampel-Modells werden mit den Ausgängen der 8-Bit-Parallel-Ausgabe verbunden.
- Die Ausgänge des Verkehrsampel-Modells werden mit den Eingängen der 8-Bit-Parallel-Eingabe verbunden.
- Die 0-V-Schiene und die +5-V-Schiene des Modells werden mit den entsprechenden Ausgängen des stabilisierten Netzgeräts verbunden.  
Alternative: Die +5-V-Schiene des Modells wird mit der +5-V-Meßbuchse des Spannungseinschubs des MFA-Mikrocomputers verbunden.
- Die Verbindungen innerhalb des Modells werden hergestellt:  
Klemme X1.3 mit Klemme X2.2,  
Klemme X3.3 mit Klemme X4.2,  
Klemme X2.1 mit Klemme X4.1,  
Klemme X4.1 mit Klemme X5.1,  
Klemme X6.1 mit +5-V-Schiene.

Die Verkabelung sollte vom Ausbilder kontrolliert werden, bevor die Geräte eingeschaltet werden.

---

**Aufgabe A1**

---

Name:

---

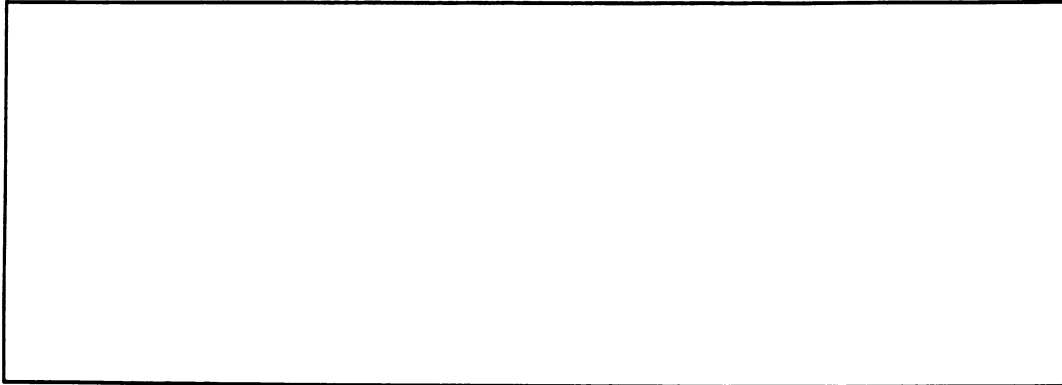
Kurs:

---

Datum:

---

**A1a:** Zeichnen Sie den Stromlaufplan für S2.



Beantworten Sie folgende Fragen:

**A1b:** Welche Funktion hat S1?

**A1c:** Welche Funktion hat S3?

**A1d:** Welche Funktion haben H1, H2 und H3?

**A1e:** Welche Funktion haben H4 und H5?

**4.2 Aufgabe A2: Erkundung der Verkehrsampeln**

Die Wirkung der Ausgänge A00 bis A05 wird erkundet. Zur Durchführung des Versuchs werden Datenwörter mit Hilfe des OUT-Kommandos des MAT 85 ausgegeben. Die Adresse der 8-Bit-Parallel-Ausgabe ist 0.

KMD>OUT

PORT-NR = 00

DATEN = 00 \_

O CR oder SP eintippen,

“UT” wird ergänzt.

CR oder SP eintippen.

Daten eingeben.

Die Daten werden in hexadezimaler Schreibweise eingegeben. Sie werden zweckmäßigerweise mit SP (space = Leertaste) und nicht mit CR (carriage return = Eingabetaste) abgeschlossen, um das Kommando nicht für jede Eingabe erneut aufrufen zu müssen. Erst nach der letzten Eingabe CR verwenden.

Im Ruhezustand (Daten = 00) sind die Lampen aller Ampeln ausgeschaltet. Um nur H6 (rote Lampe der Verkehrsampel 3) einzuschalten, muß laut Zuordnungsliste eingegeben werden:

DATEN = 00 08      08 SP eingeben (Eingabe unterstrichen).

---

**Aufgabe A2**

---

Name: 

---

---

Kurs: 

---

---

Datum: 

---

**A2a:** Um nur H3 (grüne Lampe der Verkehrsampel 1) einzuschalten, muß mit Hilfe des OUT-Kommandos ausgegeben werden:

**A2b:** Um nur H7 (gelbe Lampe der Verkehrsampel 3) einzuschalten, muß mit Hilfe des OUT-Kommandos ausgegeben werden:

**A2c:** Um H1 und H2 (rote und gelbe Lampe der Verkehrsampel 1) einzuschalten, muß mit Hilfe des OUT-Kommandos ausgegeben werden:

**4.3 Aufgabe A3: Erkundung der Fußgängerampeln**

Die Wirkung der Ausgänge, insbesondere A06 und A07, wird erkundet. Zur Durchführung des Versuchs wird das OUT-Kommando des MAT 85 benutzt. Die Adresse der 8-Bit-Parallel-Ausgabe ist 0.

KMD>OUT

PORT-NR = 00

DATEN = 00 \_

O CR oder SP eintippen,  
"UT" wird ergänzt.  
CR oder SP eintippen.  
Daten eingeben.

Die Daten werden in hexadezimaler Schreibweise eingegeben. Sie werden zweckmäßigerweise mit SP (space = Leertaste) und nicht mit CR (carriage return = Eingabetaste) abgeschlossen, um das Kommando nicht für jede Eingabe erneut aufrufen zu müssen. Erst nach der letzten Eingabe CR verwenden.

---

**Aufgabe A3**

---

Name:

---

Kurs:

---

Datum:

---

**A3a:** Welche Funktion wird durch den Ausgang A07 gesteuert?

Um nur A07 auf High zu schalten, muß eingegeben werden:

DATEN = 00 80      80 SP eingeben.

Beschreiben Sie Ihre Beobachtung bez. der Fußgängerampeln 2, 4 und 5.

**A3b:** Um die Fußgängerampel 5 auf rot zu schalten, muß welches Datenwort ausgegeben werden?

**A3c:** Wie hängen die Verkehrsampel 1 und die Fußgängerampel 2 zusammen? Um die Fußgängerampel 2 auf grün zu schalten, muß welches Datenwort ausgegeben werden?

**A3d:** Wie hängen die Verkehrsampel 3 und die Fußgängerampel 4 zusammen? Um die Fußgängerampel 4 auf grün zu schalten, muß welches Datenwort ausgegeben werden?

#### 4.4 Aufgabe A4: Erkundung der Meldeeinrichtungen

Das nachstehend aufgeführte Programm wird in den MFA-Mikrocomputer geladen. Dazu wird das Memory- oder Assembler-Kommando des MAT 85 (s. *FTÜ Kapitel 7.1*) verwendet oder das Programm von der Diskette geladen.

```
0000          ; HILFSMITTEL ZU AUFGABE A4
0000          ; ANZEIGE DER 8-BIT-PARALLEL-EINGABE (ADR. 0)
0000          ;
0000          ; BENUTZT UNTERPROG. WABIN UND WBUFI DES MAT85
0000          ;
0000      WABIN    EQU      005E      ;WRITE ACCUMULATOR BINARY,
0000                                     ;BINAERAUSGABE DES AKKU
0000      WBUFI    EQU      006D      ;WRITE BUFFER IMMEDIATE,
0000                                     ;DIREKTE TEXTAUSGABE
0000      EINGABE  EQU      0         ;8-BIT-PARALLEL-EINGABE
0000          ;
0000          ORG      0E000
E000 DB 00          START: IN      EINGABE ;LIES EINGABE
E002 CD 5E00        CALL    WABIN  ;AUSGABE
E005 CD 6D00        CALL    WBUFI  ;CURSOR ZURUECK
E008 08080808      DB      08,08,08,08,08,08,08,08,0
E00C 08080808
E010 00
E011 C3 00E0        JMP      START
```

Der Abdruck entspricht der Quelltextdatei A4.ASM, die mit dem BFZ-Editor erstellt wurde. Bei Eingabe mit dem Memory-Kommando werden nur die Hexadezimalzahlen der zweiten Spalte ab Adresse E000 eingegeben.

Bei Eingabe mit dem Assembler-Kommando werden nur die Marken in der dritten Spalte, die Operations-Codes in der vierten Spalte und die Operanden in der fünften Spalte eingegeben. Der Kommentar (die Zeichen beginnend mit dem Semikolon ";" bis zum Zeilenende) muß nicht eingegeben werden; ebenso können Zeilen, die nur Kommentar enthalten, gänzlich entfallen.

Das Programm wird mit dem Kommando GO des MAT 85 gestartet (Startadresse E000).

Die Pegel der Eingangsleitungen können an den Leuchtdioden der 8-Bit-Parallel-Eingabe abgelesen werden (High-Pegel = Leuchtdiode leuchtet = 1, Low-Pegel = Leuchtdiode dunkel = 0). Sie können außerdem an der Bildschirmanzeige abgelesen werden.

---

**Aufgabe A4**

---

Name:

---

Kurs:

---

Datum:

---

**A4a:** Bringen Sie die Schalter der Eingänge E00 bis E02 in die linke Stellung und die Schalter der Eingänge E03 bis E07 in die rechte Stellung. Alle Leuchtdioden sind dunkel. Warum?

**A4b:** Drücken Sie nur den Taster S1. Welche Leuchtdiode leuchtet? Notieren Sie die Bildschirmanzeige und unterstreichen Sie diejenige Dualstelle, die sich bei Tastendruck ändert.

**A4c:** Drücken Sie nur den Taster S2. Welche Leuchtdiode leuchtet? Notieren Sie die Bildschirmanzeige und unterstreichen Sie diejenige Dualstelle, die sich bei Tastendruck ändert.

**A4d:** Drücken Sie nur den Taster S3. Welche Leuchtdiode leuchtet? Notieren Sie die Bildschirmanzeige und unterstreichen Sie diejenige Dualstelle, die sich bei Tastendruck ändert.

**A4e:** Fassen Sie zusammen:

In Ruhestellung erzeugen die Melde-Einrichtungen ...

Wenn betätigt, erzeugen die Melde-Einrichtungen ...



**4.5 Aufgabe A5: Steuerung der Verkehrsampeln**

Es sollen die Verkehrsampeln 1 und 3 so gesteuert werden, daß wechselseitig der Verkehr aus der Hauptstraße und der Nebenstraße freigegeben wird. Die Fußgängerampeln werden nicht in Betrieb genommen.

Als Schaltzeiten werden eingesetzt:

Dauer der Grün-Phasen:	24 Sekunden,
Dauer der Gelb-Phasen:	3 Sekunden,
Dauer der Rot-Gelb-Phasen:	3 Sekunden.

Die Rot-Gelb-Phase einer Verkehrsampel schließt direkt an die Gelb-Phase der jeweils anderen Verkehrsampel an.

Die Dauer der Rot-Phasen ergibt sich aus den Phasendauern der jeweils anderen Verkehrsampel.

**Hinweis an die ausländischen Benutzer:**

Die Abfolge der Ampelphasen entspricht den Festlegungen der deutschen Straßenverkehrsordnung. In anderen Ländern können abweichende Regelungen vorliegen (z.B. Wegfall der Rot-Gelb-Phase). Die Aufgabenstellungen und Programme müssen dann sinngemäß modifiziert werden.

**Hinweis:**

Fortsetzung der Aufgabe A5

- für Assembler-Programmierung auf S. 40,
- für BASIC-Programmierung auf S. 48,
- für SPS-Programmierung auf S. 54.

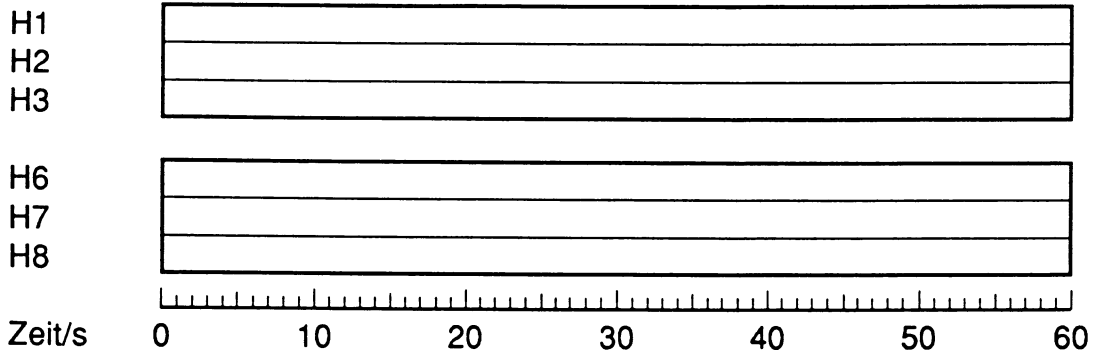
## Aufgabe A5 (Einführung)

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

**A5a:** Füllen Sie folgendes Signal-Zeit-Diagramm aus. Schraffieren Sie (vorzugsweise mit Farbstiften) die Zeitabschnitte, in denen die Lampen leuchten. Beginnen Sie mit der Rot-Gelb-Phase der Verkehrsampel 1. Schreiben Sie die Datenwörter, die zu den jeweiligen Abschnitten gehören, unter das Signal-Zeit-Diagramm. Beachten Sie, daß die Fußgängerampeln dunkel bleiben sollen.



Daten-  
wörter

**A5b:** Geben Sie die Datenwörter der Reihenfolge nach mit Hilfe des Out-Kommandos des MAT 85 aus. Kontrollieren Sie die Richtigkeit der Ampelanzeigen.

## 4.5.1 Steuerung der Verkehrsampeln (Assembler)

Um den Ablauf der Ampelsteuerung zu automatisieren, werden die Datenwörter aus A5a der Reihenfolge nach in den Akkumulator des Mikroprozessors geladen und mit dem OUT-Befehl ausgegeben. Nach jeder Ausgabe wird eine Verzögerungsschleife eingelegt, die den geforderten Schaltdauern entspricht.

Nach Abschluß der letzten Phase erfolgt ein Rücksprung an den Anfang des Programms, so daß der Ablauf sich wiederholt.

Für jede Phase ergibt sich folgende Befehlssequenz:

LXI B,00F0	Lade Registerpaar BC mit der Dauer des Zeitintervalls in Einheiten von 1/10 Sekunden. Hier: F0 (hex) = 240 (dez) = 24 Sekunden
MVI A,0C	Lade Akkumulator mit dem Datenwort für die Ausgabe.
OUT 0	Ausgabe des Akkumulator-Inhalts an die 8-Bit-Parallel-Ausgabe.
CALL DELAY	Aufruf des Unterprogramms DELAY (s.u.), das die erforderliche Verzögerung realisiert. Nach dem Ende der Ausführung des Unterprogramms DELAY geht der Programmfluß in der nächsten Zeile weiter.
LXI ...	Nächste Phase...

Zur Erzeugung der Verzögerung wird das Unterprogramm DELAY benutzt:

E03F				; UNTERPROGRAMM DELAY
E03F C5	DELAY:	PUSH	B	;REGISTER RETTEN
E040 D5		PUSH	D	
E041 F5		PUSH	PSW	
E042 11 3425	DELAY1:	LXI	D,2534	;ZEITBASIS 1/10 S
E045 1B	DELAY2:	DCX	D	;GEGEN NULL ZAEHLEN
E046 7A		MOV	A,D	;TESTE AUF NULL
E047 B3		ORA	E	
E048 C2 45E0		JNZ	DELAY2	;SCHLEIFENENDE DELAY2
E04B 0B		DCX	B	;GEGEN NULL ZAEHLEN
E04C 78		MOV	A,B	;TESTE AUF NULL
E04D B1		ORA	C	
E04E C2 42E0		JNZ	DELAY1	;SCHLEIFENENDE DELAY1
E051 F1		POP	PSW	
E052 D1		POP	D	
E053 C1		POP	B	
E054 C9		RET		;UNTERPROGRAMM DELAY

---

**Aufgabe A5 (Assembler)**  
**(Blatt 1 von 4)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

**A5c:** Füllen Sie die Leerstellen in dem folgenden Assemblerprogramm durch Konstanten aus. Die Konstanten nach dem LXI-Befehl sind die festgelegten Zeitdauern (Zwei-Byte-Konstanten) und die Konstanten nach dem MVI-Befehl die zuvor ermittelten Datenwörter. Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln entsprechend dem Signal-Zeit-Diagramm in Aufgabe A5a durchführen.

```
0000          ; AUFGABE A5C
0000          ; STEUERUNG DES FAHRZEUGVERKEHRS
0000          ;
0000          EINGABE EQU    0          ;8-BIT-PARALLEL-EINGABE
0000          AUSGABE EQU    0          ;8-BIT-PARALLEL-AUSGABE
0000          ;
0000          ORG      0E000    ;STARTADRESSE
E000          ;
E000 01 ....  PHASE1: LXI      B,....    ;3 SEKUNDEN
E003 3E ..      MVI      A,..    ;ROT1 + GELB1 + ROT3
E005 D3 00      OUT      AUSGABE
E007 CD 3FE0    CALL     DELAY
E00A 01 ....  PHASE2: LXI      B,....    ;24 SEKUNDEN
E00D 3E ..      MVI      A,..    ;GRUEN1 + ROT3
E00F D3 00      OUT      AUSGABE
E011 CD 3FE0    CALL     DELAY
E014 01 ....  PHASE3: LXI      B,....    ;3 SEKUNDEN
E017 3E ..      MVI      A,..    ;GELB1 + ROT3
E019 D3 00      OUT      AUSGABE
E01B CD 3FE0    CALL     DELAY
E01E 01 ....  PHASE4: LXI      B,....    ;3 SEKUNDEN
E021 3E ..      MVI      A,..    ;ROT1 + ROT3 + GELB3
E023 D3 00      OUT      AUSGABE
E025 CD 3FE0    CALL     DELAY
E028 01 ....  PHASE5: LXI      B,....    ;24 SEKUNDEN
E02B 3E ..      MVI      A,..    ;ROT1 + GRUEN3
E02D D3 00      OUT      AUSGABE
E02F CD 3FE0    CALL     DELAY
E032 01 ....  PHASE6: LXI      B,....    ;3 SEKUNDEN
E035 3E ..      MVI      A,..    ;ROT1 + GELB3
E037 D3 00      OUT      AUSGABE
E039 CD 3FE0    CALL     DELAY
E03C          ;
E03C C3 00E0    JMP      PHASE1    ;ENDLOSSCHLEIFE
E03F          ;
E03F          ;      UNTERPROGRAMM DELAY
E03F C5      DELAY: PUSH     B          ;REGISTER RETTEN
E040 D5      PUSH     D
E041 F5      PUSH     PSW
```

In dem Unterprogramm DELAY macht man sich zunutze, daß die Befehle für den Mikroprozessor jeweils eine fest vorgegebene Zeit zu ihrer Bearbeitung benötigen. Durch gezielte Wiederholung der Befehle wird die gewünschte Verzögerung erreicht.

**Detail-Information:**

Das Unterprogramm DELAY sichert in den ersten drei Befehlen den Inhalt aller CPU-Register, die es benutzt, auf den Stapelspeicher. Die entsprechenden Befehle, die den Registerinhalt entsprechend der Information des Stapelspeichers wieder herstellen, befinden sich am Ende des Unterprogramms, vor dem Rücksprungbefehl. Das Unterprogramm hinterläßt durch diese Technik sämtliche Register unverändert.

Das Unterprogramm besteht weiterhin aus zwei ineinander geschachtelten Schleifen. Die Zahl der Schleifendurchläufe der inneren Schleife wurde so gewählt, daß dieser Programmblock eine Ausführzeit von ca. 1/10 Sekunde benötigt:

	Operations-Code	Taktzyklen	Zeit / $\mu$ s
DELAY2:	DCX D	6	3
	MOV A,D	4	2
	ORA E	4	2
	JNZ DELAY2	7	3,5
		10	5 beim letzten Durchlauf
Summe		21	10,5
		24	12 beim letzten Durchlauf

Zahl der Durchläufe: 2534 (hex) = 9524 (dez)

Gesamtzeit:  $9523 \times 10,5 \mu\text{s} + 12 \mu\text{s} = 100\,003,5 \mu\text{s}$

Die äußere Schleife wird entsprechend dem Parameter, der in dem Registerpaar BC übergeben wird, wiederholt. Der Zeitbedarf dieser äußeren Schleife und der Stapelspeicher-Operationen wird nicht berücksichtigt. Die Gesamtzeiten sind daher geringfügig länger, als zu erwarten steht.

---

**Aufgabe A5 (Assembler)**  
**(Blatt 2 von 4)**

---

Name:

---

Kurs:

---

Datum:

---

```
E042 11 3425      DELAY1: LXI      D,2534 ;ZEITBASIS 1/10 S
E045 1B           DELAY2: DCX      D      ;GEGEN NULL ZAEHLEN
E046 7A           MOV      A,D      ;TESTE AUF NULL
E047 B3           ORA      E
E048 C2 45E0      JNZ      DELAY2   ;SCHLEIFENENDE DELAY2
E04B 0B           DCX      B      ;GEGEN NULL ZAEHLEN
E04C 78           MOV      A,B      ;TESTE AUF NULL
E04D B1           ORA      C
E04E C2 42E0      JNZ      DELAY1   ;SCHLEIFENENDE DELAY1
E051 F1           POP      PSW
E052 D1           POP      D
E053 C1           POP      B
E054 C9           RET              ;UNTERPROGRAMM DELAY
```

Geben Sie das Assemblerprogramm mit Hilfe des Memory- oder Assembler-Kommandos des MAT 85 ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des Print- oder Disassembler-Kommandos.

Starten Sie das Programm mit dem GO-Kommando.

Kontrollieren Sie den korrekten Ablauf der Ampelphasen. Kontrollieren Sie die korrekten Schaltzeiten mit Hilfe einer Stoppuhr.

**4.5.2 Steuerung mit dem Tabellenverfahren (Assembler)**

Die in allen Ampelphasen gleichbleibende Struktur läßt vermuten, daß das Programm auch kompakter verfaßt werden könnte. In der Tat unterscheiden sich die Ampelphasen nur durch die Zeitdauer und die auszugebenden Datenwörter.

Der Grundgedanke der Programmvereinfachung liegt darin, die jeweils in einer Ampelphase auftretenden Konstanten einer Tabelle zu entnehmen und in einem einzigen, für alle Ampelphasen gleichbleibenden Programmteil zu verwenden. Dieser Programmteil wird mit wechselnden Konstanten so oft wiederholt, wie Ampelphasen auszuführen sind.

Dem nebenstehenden Programm der Aufgabe A5d nach dem Tabellenverfahren liegen folgende Festlegungen zugrunde:

- Die Tabelle wird im RAM-Bereich im Anschluß an das Programm angelegt.
- Das Registerpaar HL wird beim Programmstart mit der Adresse des Tabellenanfangs geladen ( `LXI H, TAB` ). Das Registerpaar dient als sog. Zeiger.
- In den nachfolgenden Programmteilen kann auf die Tabelle mit Hilfe des Zeigers zugegriffen werden. Direkte Speicherzugriffe werden von dem Mikroprozessor 8085 praktisch nicht unterstützt. Programmtechnisch erfolgt der Speicherzugriff durch die Verwendung des Pseudoregisters M (z.B.: `MOV A, M` ). D.h.: M ist die Schreibweise für die Speicherzelle, deren Adresse im Registerpaar HL steht.
- Die Zahl der Programmdurchläufe bis zur vollständigen Abarbeitung der Tabelle wird zu Programmstart in das Register D geladen ( `MVI D, 06` ).
- Am Ende des zu wiederholenden Programmstücks wird der Inhalt des Registers D um eins erniedrigt ( `DCR D` ) und anschließend überprüft.
- Wenn der Inhalt des Registers noch nicht die Null erreicht hat, wird das Programmstück wiederholt ( `JNZ SCHL` ), anderenfalls geht es mit der nächsten Zeile weiter, die wieder zum Programmanfang zurückführt ( `JMP START` ) und somit neuerlich sechs Wiederholungen auslöst.
- In jedem Schleifendurchlauf wird zunächst das Ausgabe-Datenwort laut Aufgabe A5a eingelesen ( `MOV A, M` ), anschließend die Phasendauer. Nach jedem Einlesen wird der Zeiger HL um eins erhöht ( `INX H` ), damit er auf die nächste zu lesende Speicherzelle zeigt. Die Phasendauer wird in zwei Einlesevorgängen aus dem Speicher übernommen, da es sich hierbei um ein Zwei-Byte-Wort handelt. Dabei wird zunächst das niederwertige ( `MOV C, M` ) und dann das höherwertige Byte gelesen ( `MOV B, M` ).
- Die Ausgabe an die 8-Bit-Parallel-Ausgabe ( `OUT A, AUSGABE` ) und der Aufruf des Unterprogramms DELAY ( `CALL DELAY` ) sind unverändert.

**Aufgabe A5 (Assembler)**  
**(Blatt 3 von 4)**

Name:

Kurs:

Datum:

- ▲ **A5d:** Füllen Sie die Leerstellen im Datenbereich des folgenden Assemblerprogramms durch Konstanten aus. Die Konstanten nach den DW-Assembleranweisungen sind die festgelegten Zeitdauern (Zwei-Byte-Konstanten) und die Konstanten nach den DB-Assembleranweisungen die zuvor ermittelten Datenwörter. Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln entsprechend dem Signal-Zeit-Diagramm in Aufgabe A5a durchführen.

```
0000          ; AUFGABE A5D
0000          ; STEUERUNG DES FAHRZEUGVERKEHRS
0000          ; VARIANTE MIT DEM TABELLENVERFAHREN
0000          ;
0000          EINGABE EQU      0          ; 8-BIT-PARALLEL-EINGABE
0000          AUSGABE EQU     0          ; 8-BIT-PARALLEL-AUSGABE
0000          ;
0000          ORG      0E000      ; STARTADRESSE
E000          ;
E000 16 06      START: MVI      D,06      ; ZAHL DER PHASEN
E002 21 2DE0      LXI      H,TAB      ; ZEIGER AUF PROZESSTABELLE
E005 7E          SCHL: MOV      A,M      ; HOLE AUSGABEKONSTANTE
E006 D3 00          OUT      AUSGABE
E008 23          INX      H          ; ZEIGER WEITER
E009 4E          MOV      C,M          ; LOW-BYTE DER DAUER
E00A 23          INX      H          ; ZEIGER WEITER
E00B 46          MOV      B,M          ; HIGH-BYTE DER DAUER
E00C 23          INX      H          ; ZEIGER WEITER
E00D CD 17E0      CALL     DELAY
E010 15          DCR      D          ; PHASENZAEHLER GEGEN NULL
E011 C2 05E0      JNZ      SCHL      ; WIEDERHOLSCHLEIFE
E014 C3 00E0      JMP      START      ; GESAMTPROZESS WIEDERHOLEN
E017          ;
E017          ; UNTERPROGRAMM DELAY
E017 C5          DELAY: PUSH     B          ; REGISTER RETTEN
E018 D5          PUSH     D
E019 F5          PUSH     PSW
E01A 11 3425      DELAY1: LXI      D,2534      ; ZEITBASIS 1/10 S
E01D 1B          DELAY2: DCX      D          ; GEGEN NULL ZAEHLEN
E01E 7A          MOV      A,D          ; TESTE AUF NULL
E01F B3          ORA      E
E020 C2 1DE0      JNZ      DELAY2      ; SCHLEIFENENDE DELAY2
E023 0B          DCX      B          ; GEGEN NULL ZAEHLEN
E024 78          MOV      A,B          ; TESTE AUF NULL
E025 B1          ORA      C
E026 C2 1AE0      JNZ      DELAY1      ; SCHLEIFENENDE DELAY1
E029 F1          POP      PSW          ; REGISTER WIEDER HERSTELLEN
E02A D1          POP      D
E02B C1          POP      B
E02C C9          RET          ; UNTERPROGRAMM DELAY
E02D          ;
```



- Das Unterprogramm DELAY ist ebenfalls unverändert.
- Die Tabelle wird am Ende des Programms angelegt. Zur Ablage im Speicher dienen Assembler-Anweisungen. Mit der Anweisung DB werden Ein-Byte-Konstanten abgelegt, mit der Anweisung DW Zwei-Byte-Konstanten.

Neben der Verkürzung des Programms wird durch das Tabellenverfahren auch eine höhere Flexibilität erzielt: Bei Änderungen oder Weiterentwicklungen des Programms (siehe Aufgabe A6) muß nur noch die Tabelle und ggf. die Anzahl der Phasen angepaßt werden.

**Hinweis:**

Das Tabellenverfahren wird auch ausführlich in *FTÜ, Kapitel 20.1, Seite 44ff* diskutiert.

---

**Aufgabe A5 (Assembler)**  
**(Blatt 4 von 4)**

---

Name:

---

Kurs:

---

Datum:

---

```
E02D          ; TABELLE DER AMPELPHASEN
E02D ..      TAB:  DB      ..      ;ROT1 + GELB1 + ROT3
E02E ....    DW      ....    ;3 SEKUNDEN
E030 ..      DB      ..      ;GRUEN1 + ROT3
E031 ....    DW      ....    ;24 SEKUNDEN
E033 ..      DB      ..      ;GELB1 + ROT3
E034 ....    DW      ....    ;3 SEKUNDEN
E036 ..      DB      ..      ;ROT1 + ROT3 + GELB3
E037 ....    DW      ....    ;3 SEKUNDEN
E039 ..      DB      ..      ;ROT1 + GRUEN3
E03A ....    DW      ....    ;24 SEKUNDEN
E03C ..      DB      ..      ;ROT1 + GELB3
E03D ....    DW      ....    ;3 SEKUNDEN
E03F          ;
```

Geben Sie das Assemblerprogramm mit Hilfe des Memory- oder Assembler-Kommandos des MAT 85 ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des Print- oder Disassembler-Kommandos.

Starten Sie das Programm mit dem GO-Kommando.

Kontrollieren Sie den korrekten Ablauf der Ampelphasen.

Kontrollieren Sie die korrekten Schaltzeiten mit Hilfe einer Stoppuhr.

**4.5.3 Steuerung der Verkehrsampeln (BASIC)**

Um den Ablauf der Ampelsteuerung zu automatisieren, werden die Datenwörter aus A5a der Reihenfolge nach mit dem OUT-Befehl ausgegeben. Nach jeder Ausgabe wird eine Verzögerungsschleife eingelegt, die den geforderten Schaltdauern entspricht.

Nach Abschluß der letzten Phase erfolgt ein Rücksprung an den Anfang des Programms, so daß der Ablauf sich wiederholt.

Für jede Phase ergibt sich folgende Befehlssequenz:

<code>Z=240</code>	Besetze die Variable Z (für <u>Z</u> eit) mit der Dauer des Zeitintervalls in Einheiten von 1/10 Sekunden. Hier: 240 für 24 Sekunden
<code>OUT 0,12</code>	Ausgabe des Datenworts an die 8-Bit-Parallel-Ausgabe.
<code>GOSUB 1000</code>	Aufruf des Unterprogramms VERZOEGERUNG (s.u.), das die erforderliche Verzögerung realisiert. Nach dem Ende der Ausführung des Unterprogramms VERZOEGERUNG geht der Programmfluß in der nächsten Zeile weiter.
<code>Z= ...</code>	Nächste Phase ...

Zur Erzeugung der Verzögerung wird das Unterprogramm VERZOEGERUNG benutzt:

```
1000 REM VERZOEGERUNG (Z = ZEIT IN 1/10 SEKUNDE)
1010 FOR I=1 TO Z
1020 FOR J=1 TO 98
1030 NEXT J
1040 NEXT I
1050 RETURN
```

Hier macht man sich zunutze, daß der BASIC-Interpreter jeweils eine fest vorgegebene Zeit zu der Bearbeitung der Befehle benötigt. Durch gezielte Wiederholung der Befehle wird die gewünschte Verzögerung erreicht.

---

**Aufgabe A5 (BASIC)**  
**(Blatt 1 von 2)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

**A5c:** Füllen Sie die Leerstellen in dem folgenden BASIC-Programm durch Konstanten aus. Die Konstanten in der Zuweisung Z=... sind die festgelegten Zeitdauern und die Konstanten nach dem OUT-Befehl die zuvor ermittelten Datenwörter. Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln entsprechend dem Signal-Zeit-Diagramm in Aufgabe A5a durchführen.

```
10 REM AUFGABE A5C
20 REM STEUERUNG DES FAHRZEUGVERKEHRS
30 REM
40 A=0                                :REM 8-BIT-PARALLEL-AUSGABE
100 Z=..                              :REM 3 SEKUNDEN
110 OUT A,..                          :REM ROT1 + GELB1 + ROT3
120 GOSUB 1000
200 Z=...                             :REM 24 SEKUNDEN
210 OUT A,..                          :REM GRUEN1 + ROT3
220 GOSUB 1000
300 Z=..                              :REM 3 SEKUNDEN
310 OUT A,..                          :REM GELB1 + ROT3
320 GOSUB 1000
400 Z=..                              :REM 3 SEKUNDEN
410 OUT A,..                          :REM ROT1 + ROT3 + GELB3
420 GOSUB 1000
500 Z=...                             :REM 24 SEKUNDEN
510 OUT A,..                          :REM ROT1 + GRUEN3
520 GOSUB 1000
600 Z=..                              :REM 3 SEKUNDEN
610 OUT A,..                          :REM ROT1 + GELB3
620 GOSUB 1000
999 GOTO 100
1000 REM VERZOEGERUNG (Z = ZEIT IN 1/10 SEKUNDE)
1010 FOR I=1 TO Z
1020 FOR J=1 TO 98
1030 NEXT J
1040 NEXT I
1050 RETURN
```

Geben Sie das BASIC-Programm ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.

Starten Sie das Programm mit dem Kommando RUN.

Kontrollieren Sie den korrekten Ablauf der Ampelphasen.

Kontrollieren Sie die Schaltzeiten mit Hilfe einer Stoppuhr.

**Detail-Information:**

Das Unterprogramm VERZOEGERUNG besteht aus zwei ineinander geschachtelten Schleifen. Die Zahl der Durchläufe der inneren Schleife wurde so gewählt, daß dieser Programmblock eine Ausführzeit von ca. 1/10 Sekunde benötigt. Beim Vergleich mit dem entsprechenden Programmstück in der Assemblerversion fällt auf:

- Die Zahl der Schleifendurchläufe ist im BASIC-Programm wesentlich niedriger als im Assemblerprogramm (98 gegenüber 9524). Dies läßt darauf schließen, daß die Ausführung eines BASIC-Programmstücks etwa 100mal so lange dauert wie das entsprechende Programmstück in Assembler.

Der Hauptgrund dafür: Beim Assembler-Programm wird das Programm vorab in Maschinensprache übersetzt - von Hand bei Benutzung des Memory-Kommandos, jeweils nach der Eingabe bei Benutzung des Assembler-Kommandos bzw. durch das Kommando Control-A im BFZ-Editor. Bei einem BASIC-Programm erfolgt die Übersetzung erst während der Ausführung des Programms, und zwar nicht nur einmal, sondern bei mehrfacher Abarbeitung einer Zeile (wie hier in der Wiederholungschleife) jedesmal erneut.

- Bei dem BASIC-Programm läßt sich die Ausführzeit eines Programmabschnitts in der Praxis nicht aus der Zahl der Maschinenbefehle und deren Ausführzeit ermitteln, da zu viele Maschinenbefehle durchlaufen werden und man diese in der Regel auch nicht kennt. Die Ausführzeit muß vielmehr im Experiment ermittelt werden.

Die äußere Schleife wird entsprechend der Parameterübergabe in der Variablen Z wiederholt.

---

## 4. Aufgaben

---

**4.5.4 Steuerung mit dem Tabellenverfahren (BASIC)**

Die in allen Ampelphasen gleichbleibende Struktur läßt vermuten, daß das Programm auch kompakter verfaßt werden könnte. In der Tat unterscheiden sich die Ampelphasen nur durch die Zeitdauer und die auszugebenden Datenwörter.

Der Grundgedanke der Programmvereinfachung liegt darin, die jeweils in einer Ampelphase auftretenden Konstanten einer Tabelle zu entnehmen und in einem einzigen, für alle Ampelphasen gleichbleibenden Programmteil zu verwenden. Dieser Programmteil wird mit wechselnden Konstanten sooft wiederholt, wie Ampelphasen auszuführen sind.

Neben der Verkürzung des Programms wird durch das Tabellenverfahren auch eine höhere Flexibilität erzielt: Bei Änderungen oder Weiterentwicklungen des Programms (siehe Aufgabe A6) muß nur noch die Tabelle und ggf. die Anzahl der Phasen angepaßt werden.

Die Tabelle wird in DATA-Zeilen am Ende des Programms angelegt.

**Hinweis:**

Das Tabellenverfahren wird auch ausführlich in *FTÜ, Kapitel 20.1, Seite 79ff* diskutiert.

---

**Aufgabe A5 (BASIC)**  
**(Blatt 2 von 2)**

---

Name:

---

---

Kurs:

---

---

Datum:

---

- ▲ **A5d:** Füllen Sie die Leerstellen in den DATA-Zeilen des folgenden BASIC-Programms durch Konstanten aus. Eine DATA-Zeile steht immer für eine Ampelphase (dies ist keine Notwendigkeit, sondern dient der Übersichtlichkeit). Die erste Konstante in einer Zeile ist jeweils eines der zuvor ermittelten Datenwörter. Die zweite Konstante in einer Zeile ist die dazugehörige Phasendauer in Zehntel-Sekunden. Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln entsprechend dem Signal-Zeit-Diagramm in Aufgabe A5a durchführen.

```
10 REM AUFGABE A5D
20 REM STEUERUNG DES FAHRZEUGVERKEHRS
30 REM VARIANTE MIT DEM TABELLENVERFAHREN
40 REM
50 A=0                      :REM 8-BIT-PARALLEL-AUSGABE
100 P=6                     :REM ZAHL DER PHASEN
110 RESTORE 2010            :REM ZEIGER AUF TABELLENANFANG
120 READ B,Z                :REM AUSGABEBYTE UND ZEITDAUER
130 OUT A,B                 :REM AUSGABE
140 GOSUB 1000
150 P=P-1                   :REM PHASENNUMMER
160 IF P>0 THEN GOTO 120
170 GOTO 100                :REM GESAMTPROZESS WIEDERHOLEN
1000 REM VERZOEGERUNG (Z = ZEIT IN 1/10 SEKUNDEN)
1010 FOR I=1 TO Z
1020 FOR J=1 TO 98
1030 NEXT J
1040 NEXT I
1050 RETURN
2000 REM DATENBEREICH
2010 DATA ...,...         :REM ROT1 + GELB1 + ROT3, 3 SEK.
2020 DATA ...,...         :REM GRUEN1 + ROT3, 24 SEK.
2030 DATA ...,...         :REM GELB1 + ROT3
2040 DATA ...,...         :REM ROT1 + ROT3 + GELB3, 3 SEK.
2050 DATA ...,...         :REM ROT1 + GRUEN3, 24 SEK.
2060 DATA ...,...         :REM ROT1 + GELB3, 3 SEK.
```

Geben Sie das BASIC-Programm ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.

Starten Sie das Programm mit dem Kommando RUN.

Kontrollieren Sie den korrekten Ablauf der Ampelphasen.

Kontrollieren Sie die Schaltzeiten mit Hilfe einer Stoppuhr.



**4.5.5 Steuerung der Verkehrsampeln (SPS)**

Um den Ablauf der Ampelsteuerung zu automatisieren, wird eine Schrittkette aufgebaut, die - entsprechend den Ampelphasen - aus sechs Schritten besteht.

Der Stand der Schrittkette wird in den Merkern M00 bis M05 gespeichert. Als Anfangsbedingung wird verwendet, daß die Merker M00 bis M05 alle- samt nicht gesetzt sind; diese Bedingung wird durch den SPS-Interpreter beim Programmstart automatisch erzeugt und ist wegen obengenannter Speicherung zu keinem späteren Zeitpunkt der Programmausführung mehr gegeben. Die Anfangsbedingung wird benutzt, um Merker M00 zu setzen. Die Benutzung der Schrittmerker wird folgendermaßen festgelegt:

- M00 - Rot-Gelb-Phase der Verkehrsampel 1
- M01 - Grün-Phase der Verkehrsampel 1
- M02 - Gelb-Phase der Verkehrsampel 1
- M03 - Rot-Gelb-Phase der Verkehrsampel 3
- M04 - Grün-Phase der Verkehrsampel 3
- M05 - Gelb-Phase der Verkehrsampel 3

Weiterhin benutzt das Programm sechs Software-Timer Z00 bis Z05, von denen jeder einer Ampelphase zugeordnet ist. Mit Vorliegen der Anfangs- bedingung wird jeder Software-Timer mit der erforderlichen Phasendauer geladen (Zeiteinheit = 1/10 Sekunde). Außerdem wird der Software-Timer Z00 durch die Anfangsbedingung gestartet.

Beim Übergang von einem Schritt zum nächsten, d.h. beim Umschalten von einer Ampelphase zur nächsten, spielt sich folgendes ab:

Die Bedingung für das Ende eines Schritts ist durch das Ablaufen des entsprechenden Software-Timers gegeben. Sie bewirkt

- das Rücksetzen des abgelaufenen Software-Timers,
- das Rücksetzen des Schrittmerkers des abgelaufenen Schritts,
- das Starten des Software-Timers des jetzt beginnenden Schritts und
- das Setzen des Schrittmerkers des jetzt beginnenden Schritts.

Auf diese Weise stößt das Ende eines Schrittes den nächsten an, beim sechsten letztlich wieder den ersten, wodurch sich der Kreis schließt.

Die Anfangsbedingung zusammen mit den sechs Übergangsbedingungen bilden zusammen den Logikteil des Programms. Dieser Teil bewirkt noch keine Steuerung. Seine Arbeit kann lediglich beobachtet werden, wenn der Einschub "Zeitwerk (4-fach)" mit der Portadresse 1x (x wird nicht einge- stellt) im Baugruppenträger des MFA-Mikrocomputers eingesteckt ist. Die Leuchtdioden dieses Einschubs zeigen den Zustand der Merker M00 bis M07 an (vergl. FTÜ, Kapitel 7.2, Seite 60).

---

; AUFGABE A5C  
; STEUERUNG DES FAHRZEUGVERKEHRS  
; INITIALISIEREN DER ZEITGEBER

UN M 00 ; ANFANGSBEDINGUNG:  
UN M 01 ; ALLE MERKER NICHT  
UN M 02 ; GESETZT.  
UN M 03 ;  
UN M 04 ;  
UN M 05 ;  
=L Z 00,30 ;1. ROT-GELB-PHASE AMPEL 1  
=L Z 01,240 ;2. GRUEN-PHASE AMPEL 1  
=L Z 02,30 ;3. GELB-PHASE AMPEL 1  
=L Z 03,30 ;4. ROT-GELB-PHASE AMPEL 3  
=L Z 04,240 ;5. GRUEN-PHASE AMPEL 3  
=L Z 05,30 ;6. GELB-PHASE AMPEL 3  
=S M 00 ;  
=S Z 00 ;1 STARTE PHASE 1

; ABLAUFKETTE (GESTEUERT DURCH ZEITGEBER)

U Z 00 ; PHASE 1 ABGELAUFEN  
=R Z 00 ;  
=R M 00 ;  
=S Z 01 ; STARTE PHASE 2  
=S M 01 ;2

U Z 01 ; PHASE 2 ABGELAUFEN  
=R Z 01 ;  
=R M 01 ;  
=S Z 02 ; STARTE PHASE 3  
=S M 02 ;3

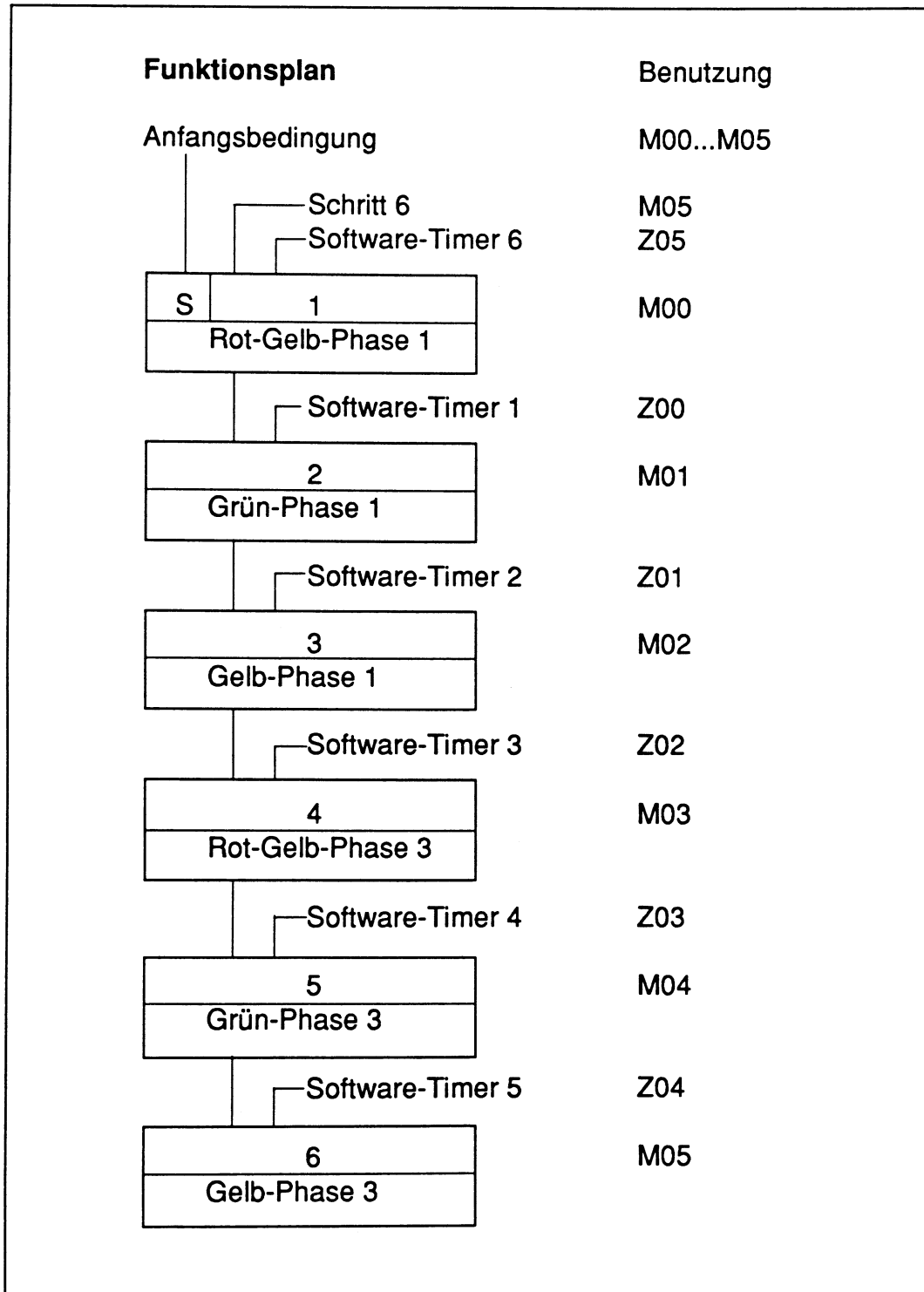
U Z 02 ; PHASE 3 ABGELAUFEN  
=R Z 02 ;  
=R M 02 ;  
=S Z 03 ; STARTE PHASE 4  
=S M 03 ;4

U Z 03 ; PHASE 4 ABGELAUFEN  
=R Z 03 ;  
=R M 03 ;  
=S Z 04 ; STARTE PHASE 5  
=S M 04 ;5

U Z 04 ; PHASE 5 ABGELAUFEN  
=R Z 04 ;  
=R M 04 ;  
=S Z 05 ; STARTE PHASE 6  
=S M 05 ;6

U Z 05 ; PHASE 6 ABGELAUFEN  
=R Z 05 ;  
=R M 05 ;  
=S Z 00 ; STARTE PHASE 1  
=S M 00 ;7

Der Logikteil des Steuerungsprogramms kann besonders übersichtlich im Funktionsplan dargestellt werden:



An den Logikteil des Programms schließt sich der Ausgabeteil an. Im Ausgabeteil wird für jede Ausgabeleitung die Bedingung genannt, unter der sie aktiviert wird.

Beispiel:

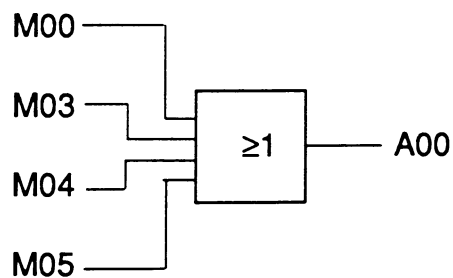
Ausgang A00 (H1, rote Lampe der Verkehrsampel 1) wird aktiviert

- in der Rot-Gelb-Phase der Verkehrsampel 1,
- in der Rot-Gelb-Phase der Verkehrsampel 3,
- in der Grün-Phase der Verkehrsampel 3,
- in der Gelb-Phase der Verkehrsampel 3.

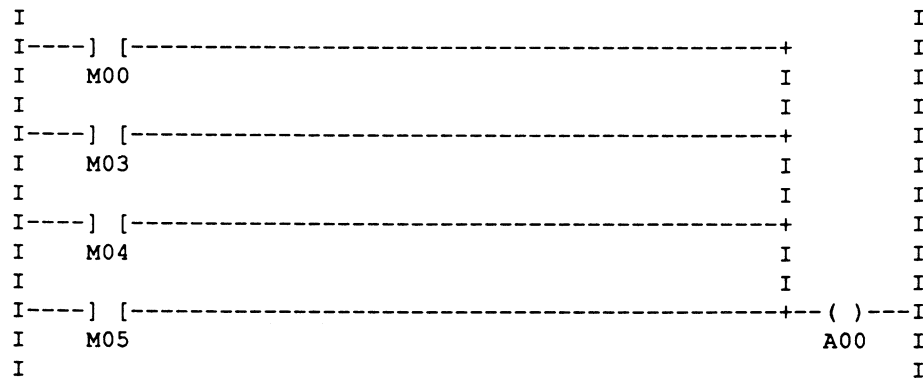
Die Phasen werden anhand der Schrittmerker ermittelt:

Sprachlich	Anweisungsliste	
	BFZ-SPS (DIN-Variante)	BFZ-SPS (laut FTÜ)
Wenn M00 oder M03 oder M04 oder M05, dann A00	UM00 OM03 OM04 OM05 =A00	*M00 +M03 +M04 +M05 =A00

### Funktionsplan



## Kontaktplan



### Hinweise:

Wenn der Kursteilnehmer die Dokumentationsfunktion des SPS-Interpreters im MAT 32K benutzt, so kann das abgedruckte Programm unverändert im BFZ-Editor eingegeben und nach Belieben mit Kommentaren versehen werden.

**Wenn der Kursteilnehmer den BFZ-SPS-Interpreter in der DIN-Variante benutzt, werden alle Kommentare und alle Leerzeichen in den Anweisungen weggelassen.**

Wenn der Kursteilnehmer den BFZ-SPS-Interpreter in der Version der FTÜ benutzt, so wird von nachstehendem gleichwertigen Programm ausgegangen:

```

*/M00*/M01*/M02*/M03*/M04*/M05=LZ00,30=LZ01,240
=LZ02,30=LZ03,30=LZ04,240=LZ05,30=SM00=SZ00
*Z00=RZ00=RM00=SZ01=SM01
*Z01=RZ01=RM01=SZ02=SM02
*Z02=RZ02=RM02=SZ03=SM03
*Z03=RZ03=RM03=SZ04=SM04
*Z04=RZ04=RM04=SZ05=SM05
*Z05=RZ05=RM05=SZ00=SM00
*M00+M03+M04+M05=A00
.....=A01
....=A02
.....=A03
.....=A04
....=A05

```

---

**Aufgabe A5 (SPS)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

**A5c:** Füllen Sie die Leerstellen in dem Ausgabeteil des folgenden SPS-Programms aus. Orientieren Sie sich dazu an dem Signal-Zeit-Diagramm der Aufgabe A5a. Die Zuordnung der logischen Verknüpfungen zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln entsprechend dem Signal-Zeit-Diagramm in Aufgabe A5a durchführen.

```
; AUFGABE A5C
; STEUERUNG DES FAHRZEUGVERKEHRS

; INITIALISIEREN DER ZEITGEBER          (siehe Seite 55)

; ABLAUFKETTE (GESTEUERT DURCH ZEITGEBER) (siehe Seite 55)

; AUSGABESTEuerung

U  M 00      ;   IN  PHASE 1
O  M 03      ;   UND PHASE 4
O  M 04      ;   UND PHASE 5
O  M 05      ;   UND PHASE 6
=  A 00      ;8  ROT1  SCHALTEN

.   .   .   ;   IN  PHASE 1
.   .   .   ;   UND PHASE 3
=  A 01      ;9  GELB1 SCHALTEN

.   .   .   ;   IN  PHASE 2
=  A 02      ;10 GRUEN1 SCHALTEN

.   .   .   ;   IN  PHASE 1
.   .   .   ;   UND PHASE 2
.   .   .   ;   UND PHASE 3
.   .   .   ;   UND PHASE 4
=  A 03      ;11 ROT3  SCHALTEN

.   .   .   ;   IN  PHASE 4
.   .   .   ;   UND PHASE 6
=  A 04      ;12 GELB3 SCHALTEN

.   .   .   ;   IN  PHASE 5
=  A 05      ;13 GRUEN3 SCHALTEN

END
```

Geben Sie das vollständige SPS-Programm (mit Initialisierung und Ablaufkette von *Seite 55*) ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.

Starten Sie das Programm mit dem Kommando GO.

Kontrollieren Sie den korrekten Ablauf der Ampelphasen.

Kontrollieren Sie die Schaltzeiten mit Hilfe einer Stoppuhr.

**4.6 Aufgabe A6: Steuerung der Verkehrs- und Fußgängerampeln**

Es sollen die Verkehrsampeln 1 und 3 sowie die Fußgängerampel 5 so gesteuert werden, daß im Uhrzeigersinn ( 1 - 3 - 5 ) der Verkehr freigegeben wird.

Bezüglich der beiden Verkehrsampeln ergeben sich keine Änderungen gegenüber Aufgabe A5.

Bei der Fußgängerampel ist folgendes zu beachten:

- Der Ausgang A07 muß auf High-Pegel gelegt werden, um die Fußgängerampeln zu aktivieren. Beachten Sie, daß damit auch die Fußgängerampeln 2 und 4 aktiviert werden. Deren Schaltung erfolgt jedoch gemäß der Verkabelung durch die Verkehrsampeln 1 und 3 (s. *Kapitel 3.4* ).
- Die Steuerung der Fußgängerampel 5 erfolgt durch den Ausgang A06.
- Auch bei der Fußgängerampel sind Übergangsphasen zu berücksichtigen, um Gelegenheit zu geben, den Übergang zu räumen. Da Fußgängerampeln jedoch keine gelbe Lampe besitzen, werden entsprechend der Rot-Gelb-Phase bzw. der Gelb-Phase der Verkehrsampeln zusätzliche Rot-Phasen eingelegt.

Als Schaltzeiten der Fußgängerampel werden festgelegt:

Dauer der Grün-Phase:	16 Sekunden,
Dauer der Übergangsphasen:	3 Sekunden.

**Hinweis:**

Fortsetzung der Aufgabe A6

- für Assembler-Programmierung auf S. 62,
- für BASIC-Programmierung auf S. 70,
- für SPS-Programmierung auf S. 74.

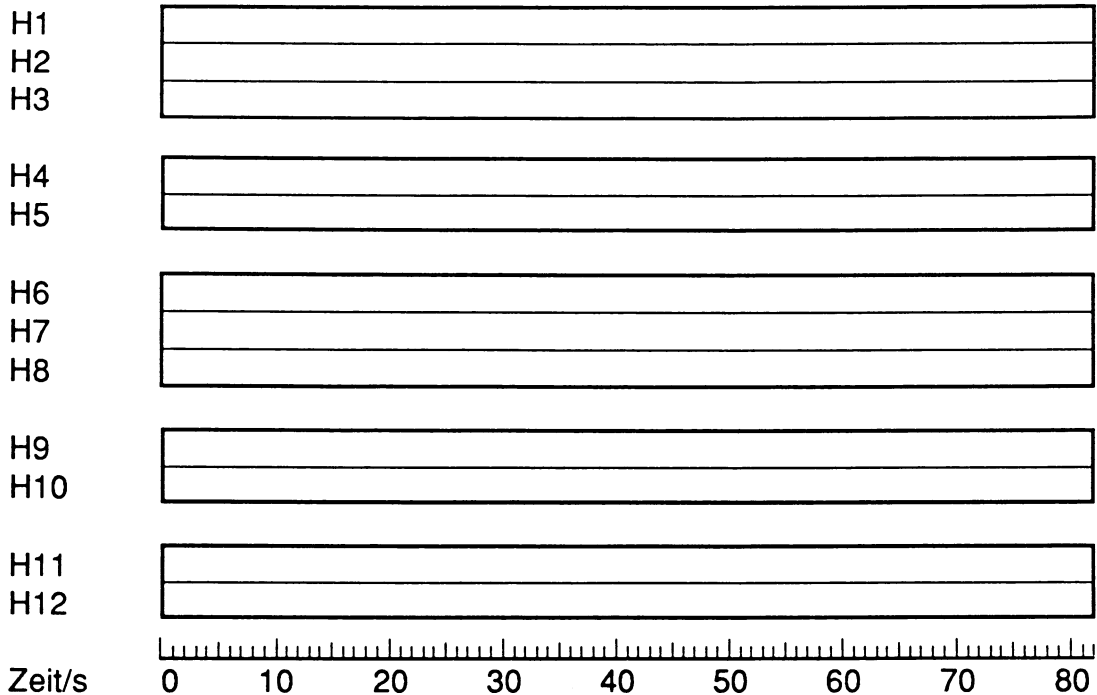
## Aufgabe A6 (Einführung)

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

**A6a:** Füllen Sie folgendes Signal-Zeit-Diagramm aus. Schraffieren Sie (vorzugsweise mit Farbstiften) die Zeitabschnitte, in denen die Lampen leuchten. Beginnen Sie mit der Rot-Gelb-Phase der Verkehrsampel 1. Schreiben Sie die Datenwörter, die zu den jeweiligen Abschnitten gehören, unter das Signal-Zeit-Diagramm.



Daten-  
wörter

**A6b:** Geben Sie die Datenwörter der Reihenfolge nach mit Hilfe des Out-Kommandos des MAT 85 aus. Kontrollieren Sie die Richtigkeit der Ampelsteuerung.



**4.6.1 Steuerung aller Ampeln (Assembler)**

Um den Ablauf der Ampelsteuerung zu automatisieren, werden die Datenwörter der Reihenfolge nach in den Akkumulator des Mikroprozessors geladen und mit dem OUT-Befehl ausgegeben. Nach jeder Ausgabe wird eine Verzögerungsschleife eingelegt, die den geforderten Schaltdauern entspricht.

Nach Abschluß des letzten Zeitintervalls erfolgt ein Rücksprung an den Anfang des Programms, so daß der Ablauf sich wiederholt.

Zur Erzeugung der Verzögerung wird wieder das Unterprogramm DELAY (siehe *Abschnitt 4.5.1* ) benutzt.

**Hinweis:**

Es bietet sich an, die Programme der Aufgabe A6 aus den Programmen der Aufgabe A5 weiterzuentwickeln. Bei den Assemblerprogrammen ist der Einsatz des BFZ-Editors vorteilhaft.

**Hilfestellung:**

Die Steuerung der Fußgängerampel 5 wird in strenger Analogie zu der der Verkehrsampeln programmiert.

---

**Aufgabe A6 (Assembler)**  
**(Blatt 1 von 4)**

---

Name:

---

Kurs:

---

Datum:

---

**A6c:** Füllen Sie die Leerzeilen in dem folgenden Assemblerprogramm mit Befehlen aus. Benutzen Sie den LXI-Befehl zum Laden der festgelegten Zeitdauern (Zwei-Byte-Konstanten) und den MVI-Befehl zum Laden der erforderlichen Datenwörter (Ein-Byte-Konstanten). Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln und Fußgängerampeln entsprechend dem Signal-Zeit-Diagramm aus Aufgabe A6a durchführen.

```
0000          ; AUFGABE A6C
0000          ; STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
0000          ;
0000          EINGABE EQU    0          ; 8-BIT-PARALLEL-EINGABE
0000          AUSGABE EQU    0          ; 8-BIT-PARALLEL-AUSGABE
0000          ;
0000          ORG      0E000      ; STARTADRESSE
E000          ;
E000 01 1E00   PHASE1: LXI      B,001E  ; 3 SEKUNDEN
E003 3E 8B     MVI      A,8B      ; ROT1 + GELB1 + ROT3
E005 D3 00     OUT      AUSGABE
E007 CD 5DE0   CALL     DELAY
E00A 01 F000   PHASE2: LXI      B,00F0  ; 24 SEKUNDEN
E00D 3E 8C     MVI      A,8C      ; GRUEN1 + ROT3
E00F D3 00     OUT      AUSGABE
E011 CD 5DE0   CALL     DELAY
E014 01 1E00   PHASE3: LXI      B,001E  ; 3 SEKUNDEN
E017 3E 8A     MVI      A,8A      ; GELB1 + ROT3
E019 D3 00     OUT      AUSGABE
E01B CD 5DE0   CALL     DELAY
E01E 01 1E00   PHASE4: LXI      B,001E  ; 3 SEKUNDEN
E021 3E 99     MVI      A,99      ; ROT1 + ROT3 + GELB3
E023 D3 00     OUT      AUSGABE
E025 CD 5DE0   CALL     DELAY
E028 01 F000   PHASE5: LXI      B,00F0  ; 24 SEKUNDEN
E02B 3E A1     MVI      A,0A1     ; ROT1 + GRUEN3
E02D D3 00     OUT      AUSGABE
E02F CD 5DE0   CALL     DELAY
E032 01 1E00   PHASE6: LXI      B,001E  ; 3 SEKUNDEN
E035 3E 91     MVI      A,91      ; ROT1 + GELB3
E037 D3 00     OUT      AUSGABE
E039 CD 5DE0   CALL     DELAY
E03C .. ....   PHASE7: ...      ..... ; 3 SEKUNDEN
E03F .. ..     ...      ..... ; ROT1 + ROT3
E041 .. ..     ...      .....
E043 .. ....   ....      .....
E046 .. ....   PHASE8: ...      ..... ; 16 SEKUNDEN
E049 .. ..     ...      ..... ; ROT1 + ROT3 + GRUEN5
E04B .. ..     ...      .....
E04D .. ....   ....      .....
```

---

## 4. Aufgaben

---

---

**Aufgabe A6 (Assembler)**  
**(Blatt 2 von 4)**Name: 

---

Kurs: 

---

Datum: 

---

```
E050 .. .... PHASE9: ... ..... ;3 SEKUNDEN
E053 .. ..      ... ..... ;ROT1 + ROT3
E055 .. ..      ... .....
E057 .. ....      .... .....
E05A           ;
E05A C3 00E0      JMP     PHASE1 ;ENDLOSSCHLEIFE
E05D           ;
E05D           ;      UNTERPROGRAMM DELAY
E05D C5      DELAY: PUSH    B      ;REGISTER RETTEN
E05E D5              PUSH    D
E05F F5              PUSH    PSW
E060 11 3425      DELAY1: LXI    D,2534 ;ZEITBASIS 1/10 S
E063 1B      DELAY2: DCX     D      ;GEGEN NULL ZAEHLEN
E064 7A              MOV     A,D    ;TESTE AUF NULL
E065 B3              ORA     E
E066 C2 63E0      JNZ     DELAY2 ;SCHLEIFENENDE DELAY2
E069 0B              DCX     B      ;GEGEN NULL ZAEHLEN
E06A 78              MOV     A,B    ;TESTE AUF NULL
E06B B1              ORA     C
E06C C2 60E0      JNZ     DELAY1 ;SCHLEIFENENDE DELAY1
E06F F1              POP     PSW
E070 D1              POP     D
E071 C1              POP     B
E072 C9              RET           ;UNTERPROGRAMM DELAY
```

Geben Sie das Programm mit Hilfe des Memory- oder Assembler-Kommandos des MAT 85 ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des Print- oder Disassembler-Kommandos.

Starten Sie das Programm mit dem GO-Kommando.

Kontrollieren Sie den korrekten Ablauf der Ampelphasen.

Kontrollieren Sie die Schaltzeiten mit Hilfe einer Stoppuhr.

**4.6.2 Erweiterung der Steuerungstabelle (Assembler)**

Die in allen Ampelphasen gleichbleibende Struktur läßt eine Erweiterung des Tabellenverfahrens aus Aufgabe A5d zu.

Folgende Änderungen müssen an dem Programm der Aufgabe A5d vorgenommen werden, um es der neuen Aufgabenstellung anzupassen:

- Die Zahl der Schleifendurchläufe muß der nun größeren Zahl der Ampelphasen angepaßt werden. Statt bisher:

MVI D,06

muß nun eingesetzt werden:

MVI D,09

- Die Datenwörter der Datentabelle ändern sich, da nun die Fußgängerampeln eingeschaltet sind.
- Der Datentabelle am Ende des Programms werden drei Zeilenpaare angefügt, die jeweils die Datenwörter und die Phasendauern für die Phasen der Fußgängerampel enthalten.

Es bietet sich an, den Aufwand bei der Entwicklung der Programme zu vergleichen. Bei der Entwicklung des Programms der Aufgabe A6c aus dem Programm der Aufgabe A5c wurde der komplette Programmcode für drei weitere Phasen ergänzt. Bei der Entwicklung des Programms der vorliegenden Aufgabe A6d aus jenem der Aufgabe A5d war weniger Aufwand notwendig. Insbesondere wurde das Programm nur um die zusätzlich benötigten Konstanten länger.

Bei umfangreichen Steuerungsprogrammen kann dieser Vorteil eine erhebliche Einsparung an Speicherplatz und einen nicht zu unterschätzenden Gewinn an Flexibilität bei der Programmentwicklung bedeuten.

---

**Aufgabe A6 (Assembler)**  
**(Blatt 3 von 4)**Name: 

---

Kurs: 

---

Datum: 

---

- ▲ **A6d:** Füllen Sie die Leerstellen in dem Datenbereich des folgenden Assemblerprogramms durch Konstanten aus. Die Konstanten nach den DW-Assembleranweisungen sind die festgelegten Zeitdauern (Zwei-Byte-Konstanten) und die Konstanten nach den DB-Assembleranweisungen die zuvor ermittelten Datenwörter. Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrs- und Fußgängerampeln entsprechend dem Signal-Zeit-Diagramm aus Aufgabe A6a durchführen.

```
0000          ; AUFGABE A6D
0000          ; STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
0000          ; VARIANTE MIT DEM TABELLENVERFAHREN
0000          ;
0000      EINGABE EQU    0          ; 8-BIT-PARALLEL-EINGABE
0000      AUSGABE EQU   0          ; 8-BIT-PARALLEL-AUSGABE
0000          ;
0000          ORG      0E000        ; STARTADRESSE
E000          ;
E000 16 09      START: MVI        D,09      ; ZAHL DER PHASEN
E002 21 2DE0      LXI        H,TAB      ; ZEIGER AUF PROZESSTABELLE
E005 7E          SCHL:  MOV        A,M      ; HOLE AUSGABEKONSTANTE
E006 D3 00      OUT        AUSGABE
E008 23          INX        H          ; ZEIGER WEITER
E009 4E          MOV        C,M      ; HIGH-BYTE DER DAUER
E00A 23          INX        H          ; ZEIGER WEITER
E00B 46          MOV        B,M      ; LOW-BYTE DER DAUER
E00C 23          INX        H          ; ZEIGER WEITER
E00D CD 17E0      CALL        DELAY
E010 15          DCR        D          ; PHASENZAEHLER GEGEN NULL
E011 C2 05E0      JNZ        SCHL      ; WIEDERHOLSCHLEIFE
E014 C3 00E0      JMP        START      ; GESAMTPROZESS WIEDERHOLEN
E017          ;
E017          ; UNTERPROGRAMM DELAY
E017 C5      DELAY:  PUSH        B          ; REGISTER RETTEN
E018 D5          PUSH        D
E019 F5          PUSH        PSW
E01A 11 3425      DELAY1: LXI        D,2534 ; ZEITBASIS 1/10 S
E01D 1B      DELAY2: DCX        D          ; GEGEN NULL ZAEHLEN
E01E 7A          MOV        A,D      ; TESTE AUF NULL
E01F B3          ORA        E
E020 C2 1DE0      JNZ        DELAY2      ; SCHLEIFENENDE DELAY2
E023 0B          DCX        B          ; GEGEN NULL ZAEHLEN
E024 78          MOV        A,B      ; TESTE AUF NULL
E025 B1          ORA        C
E026 C2 1AE0      JNZ        DELAY1      ; SCHLEIFENENDE DELAY1
E029 F1          POP        PSW      ; REGISTER WIEDER HERSTELLEN
E02A D1          POP        D
E02B C1          POP        B
E02C C9          RET              ; UNTERPROGRAMM DELAY
E02D          ;
```

---

#### 4. Aufgaben

---

---

**Aufgabe A6 (Assembler)**  
**(Blatt 4 von 4)**

---

Name:

---

Kurs:

---

Datum:

---

```
E02D          ; TABELLE DER AMPELPHASEN
E02D 8B      TAB:  DB      8B      ;ROT1 + GELB1 + ROT3
E02E 1E00    DW      001E    ;3 SEKUNDEN
E030 8C      DB      8C      ;GRUEN1 + ROT3
E031 F000    DW      00F0    ;24 SEKUNDEN
E033 8A      DB      8A      ;GELB1 + ROT3
E034 1E00    DW      001E    ;3 SEKUNDEN
E036 99      DB      99      ;ROT1 + ROT3 + GELB3
E037 1E00    DW      001E    ;3 SEKUNDEN
E039 A1      DB      0A1     ;ROT1 + GRUEN3
E03A F000    DW      00F0    ;24 SEKUNDEN
E03C 91      DB      91      ;ROT1 + GELB3
E03D 1E00    DW      001E    ;3 SEKUNDEN
E03F ..      DB      ..      ;ROT1 + ROT3
E040 ....    DW      ....    ;3 SEKUNDEN
E042 ..      DB      ...     ;ROT1 + ROT3 + GRUEN5
E043 ....    DW      ....    ;16 SEKUNDEN
E045 ..      DB      ..      ;ROT1 + ROT3
E046 ....    DW      ....    ;3 SEKUNDEN
E048          ;
```

Geben Sie das Assemblerprogramm mit Hilfe des Memory- oder Assembler-Kommandos des MAT 85 ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des Print- oder Disassembler-Kommandos.

Starten Sie das Programm.

Kontrollieren Sie den korrekten Ablauf der Ampelphasen.

Kontrollieren Sie die Schaltzeiten mit Hilfe einer Stoppuhr.



**4.6.3 Steuerung aller Ampeln (BASIC)**

Um den Ablauf der Ampelsteuerung zu automatisieren, werden die Datenwörter aus A6a der Reihenfolge nach mit dem OUT-Befehl ausgegeben. Nach jeder Ausgabe wird eine Verzögerungsschleife eingelegt, die den geforderten Schaltdauern entspricht.

Nach Abschluß des letzten Zeitintervalls erfolgt ein Rücksprung an den Anfang des Programms, so daß der Ablauf sich wiederholt.

Zur Erzeugung der Verzögerung wird wieder das Unterprogramm VERZÖGERUNG (siehe *Abschnitt 4.5.3*) benutzt.

**Hinweis:**

Es bietet sich an, die Programme der Aufgabe A6 aus den Programmen der Aufgabe A5 weiterzuentwickeln.

Bei den BASIC-Programmen wurden in Aufgabe A5 die Zeilennummern bereits so gewählt, daß die zusätzlichen Zeilen ohne Umnummerierung eingeschoben werden können.

**Hilfestellung:**

Die Steuerung der Fußgängerampel 5 wird in strenger Analogie zu der der Verkehrsampeln programmiert.

---

**Aufgabe A6 (BASIC)**  
**(Blatt 1 von 2)**

---

Name: 

---

---

Kurs: 

---

---

Datum: 

---

**A6c:** Füllen Sie die Leerstellen in den Befehlen des folgenden BASIC-Programms aus. Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln und Fußgängerampeln entsprechend dem Signal-Zeit-Diagramm aus Aufgabe A6a durchführen.

```
10 REM AUFGABE A6C
20 REM STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
30 REM
40 A=0                                :REM 8-BIT-PARALLEL-AUSGABE
100 Z=30                              :REM 3 SEKUNDEN
110 OUT A,139                         :REM ROT1 + GELB1 + ROT3
120 GOSUB 1000
200 Z=240                             :REM 24 SEKUNDEN
210 OUT A,140                         :REM GRUEN1 + ROT3
220 GOSUB 1000
300 Z=30                              :REM 3 SEKUNDEN
310 OUT A,138                         :REM GELB1 + ROT3
320 GOSUB 1000
400 Z=30                              :REM 3 SEKUNDEN
410 OUT A,153                         :REM ROT1 + ROT3 + GELB3
420 GOSUB 1000
500 Z=240                             :REM 24 SEKUNDEN
510 OUT A,161                         :REM ROT1 + GRUEN3
520 GOSUB 1000
600 Z=30                              :REM 3 SEKUNDEN
610 OUT A,145                         :REM ROT1 + GELB3
620 GOSUB 1000
700 ....                             :REM 3 SEKUNDEN
710 ... .....                        :REM ROT1 + ROT3
720 ..... .....
800 .....                             :REM 16 SEKUNDEN
810 ... .....                        :REM ROT1 + ROT3 + GRUEN5
820 ..... .....
900 ....                             :REM 3 SEKUNDEN
910 ... .....                        :REM ROT1 + ROT3
920 ..... .....
999 GOTO 100
1000 REM VERZOEGERUNG (Z = ZEIT IN 1/10 SEKUNDEN)
1010 FOR I=1 TO Z
1020 FOR J=1 TO 98
1030 NEXT J
1040 NEXT I
1050 RETURN
```

Geben Sie das BASIC-Programm ein.  
Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.  
Starten Sie das Programm mit dem Kommando RUN.  
Kontrollieren Sie den korrekten Ablauf der Ampelphasen.  
Kontrollieren Sie die Schaltzeiten mit Hilfe einer Stoppuhr.

**4.6.4 Erweiterung der Steuerungstabelle (BASIC)**

Die in allen Ampelphasen gleichbleibende Struktur läßt eine Erweiterung des Tabellenverfahrens aus Aufgabe A5d zu.

Folgende Änderungen müssen an dem Programm der Aufgabe A5d vorgenommen werden, um es der neuen Aufgabenstellung anzupassen:

- Die Zahl der Schleifendurchläufe muß der nun größeren Zahl der Ampelphasen angepaßt werden. Statt bisher:

$P=6$

muß nun eingesetzt werden:

$P=9$

- Die Datenwörter der Datentabelle ändern sich, da nun die Fußgängerampeln eingeschaltet sind.
- Der DATA-Tabelle am Ende des Programms werden drei Zeilen angefügt, die jeweils die Datenwörter und die Phasendauern für die Phasen der Fußgängerampel enthalten.

Es bietet sich an, den Aufwand bei der Entwicklung der Programme zu vergleichen. Bei der Entwicklung des Programms der Aufgabe A6c aus dem Programm der Aufgabe A5c wurde der komplette Programmcode für drei weitere Phasen ergänzt. Bei der Entwicklung des Programms der vorliegenden Aufgabe A6d aus jenem der Aufgabe A5d war weniger Aufwand notwendig. Insbesondere wurde das Programm nur um die zusätzlich benötigten DATA-Zeilen länger.

Bei umfangreichen Steuerungsprogrammen kann dieser Vorteil eine erhebliche Einsparung an Speicherplatz und einen nicht zu unterschätzenden Gewinn an Flexibilität bei der Programmentwicklung bedeuten.

---

**Aufgabe A6 (BASIC)**  
**(Blatt 2 von 2)**

---

Name:

---

Kurs:

---

Datum:

- ▲ **A6d:** Füllen Sie die Leerstellen in den DATA-Zeilen des folgenden BASIC-Programms durch Konstanten aus. Die erste Konstante in einer DATA-Zeile ist jeweils eines der zuvor ermittelten Datenwörter. Die zweite Konstante in einer Zeile ist die dazugehörige Phasendauer in 1/10-Sekunden. Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln entsprechend dem Signal-Zeit-Diagramm aus Aufgabe A6a durchführen.

```
10 REM AUFGABE A6D
20 REM STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
30 REM VARIANTE MIT DEM TABELLENVERFAHREN
40 REM
50 A=0                                :REM 8-BIT-PARALLEL-AUSGABE
100 P=9                               :REM ZAHL DER PHASEN
110 RESTORE 2010                     :REM ZEIGER AUF TABELLENANFANG
120 READ B,Z                         :REM AUSGABEBYTE UND ZEITDAUER
130 OUT A,B                          :REM AUSGABE
140 GOSUB 1000
150 P=P-1                            :REM PHASENNUMMER
160 IF P>0 THEN GOTO 120
170 GOTO 100                         :REM GESAMTPROZESS WIEDERHOLEN
1000 REM VERZOEGERUNG (Z = ZEIT IN 1/10 SEKUNDEN)
1010 FOR I=1 TO Z
1020 FOR J=1 TO 98
1030 NEXT J
1040 NEXT I
1050 RETURN
2000 REM DATENBEREICH
2010 DATA 139,30                    :REM ROT1 + GELB1 + ROT3, 3 SEK.
2020 DATA 140,240                  :REM GRUEN1 + ROT3, 24 SEK.
2030 DATA 138,30                   :REM GELB1 + ROT3, 3 SEK.
2040 DATA 153,30                   :REM ROT1 + ROT3 + GELB3, 3 SEK.
2050 DATA 161,240                  :REM ROT1 + GRUEN3, 24 SEK.
2060 DATA 145,30                   :REM ROT1 + GELB3, 3 SEK.
2070 DATA .....                   :REM ROT1 + ROT3, 3 SEK.
2080 DATA .....                   :REM ROT1 + ROT3 + GRUEN5, 16 SEK.
2090 DATA .....                   :REM ROT1 + ROT3, 3 SEK.
```

Geben Sie das BASIC-Programm ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.

Starten Sie das Programm mit dem Kommando RUN.

Kontrollieren Sie den korrekten Ablauf der Ampelphasen.

Kontrollieren Sie die Schaltzeiten mit Hilfe einer Stoppuhr.

**4.6.5 Steuerung aller Ampeln (SPS)**

Um den Ablauf der Ampelsteuerung zu automatisieren, wird eine Schrittkette aufgebaut, die - entsprechend den Ampelphasen - aus neun Schritten besteht.

Der Stand der Schrittkette wird in den Merkern M00 bis M07 und M10 gespeichert. Als Anfangsbedingung wird verwendet, daß die Merker M00 bis M10 allesamt nicht gesetzt sind; diese Bedingung wird durch den SPS-Interpreter beim Programmstart erzeugt und ist zu keinem späteren Zeitpunkt der Programmausführung mehr gegeben. Die Anfangsbedingung wird benutzt, um Merker M00 zu setzen. Die Benutzung der Schrittmerker wird folgendermaßen festgelegt:

- M00 - Rot-Gelb-Phase der Verkehrsampel 1
- M01 - Grün-Phase der Verkehrsampel 1
- M02 - Gelb-Phase der Verkehrsampel 1
- M03 - Rot-Gelb-Phase der Verkehrsampel 3
- M04 - Grün-Phase der Verkehrsampel 3
- M05 - Gelb-Phase der Verkehrsampel 3
- M06 - Übergangsphase 1 der Fußgängerampel 5
- M07 - Grün-Phase der Fußgängerampel 5
- M10 - Übergangsphase 2 der Fußgängerampel 5

Außerdem benutzt das Programm neun Software-Timer Z00 bis Z07 und Z10, von denen jeder einer Ampelphase zugeordnet ist. Mit der Anfangsbedingung wird jeder Software-Timer mit der erforderlichen Phasendauer geladen (Zeiteinheit = 1/10 Sekunde). Außerdem wird der Software-Timer Z00 mit der Anfangsbedingung gestartet.

**Hinweis:**

Es bietet sich an, die Programme der Aufgabe A6 aus den Programmen der Aufgabe A5 weiterzuentwickeln. Bei den SPS-Programmen (soweit die Dokumentationsfunktion des MAT 32K verfügbar ist) ist der Einsatz des BFZ-Editors vorteilhaft.

**Hilfestellung:**

Die Steuerung der Fußgängerampel 5 wird in strenger Analogie zu der der Verkehrsampeln programmiert.

---

**Aufgabe A6 (SPS)**  
**(Blatt 1 von 3)**

---

Name: \_\_\_\_\_

---

Kurs: \_\_\_\_\_

---

Datum: \_\_\_\_\_

---

**A6c:** Zeichnen Sie den Funktionsplan der Ablaufkette. Geben Sie auch die benutzten Schrittmerker und Software-Timer an.

Funktionsplan	Benutzung

An den Logikteil des Programms schließt sich der Ausgabeteil an. Im Ausgabeteil wird für jede Ausgabeleitung die Bedingung genannt, unter der sie aktiviert wird.

Beispiel:

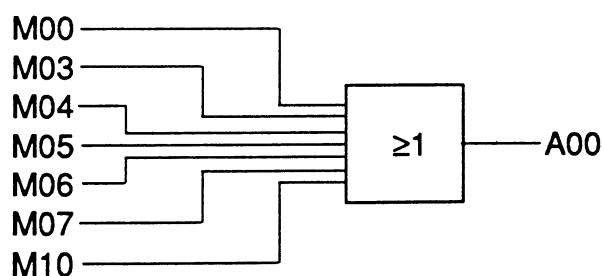
Ausgang A00 (H1, rote Lampe der Verkehrsampel 1) wird aktiviert

- in der Rot-Gelb-Phase der Verkehrsampel 1,
- in der Rot-Gelb-Phase der Verkehrsampel 3,
- in der Grün-Phase der Verkehrsampel 3,
- in der Gelb-Phase der Verkehrsampel 3,
- in der Übergangsphase 1 der Fußgängerampel 5,
- in der Grün-Phase der Fußgängerampel 5,
- in der Übergangsphase 2 der Fußgängerampel 5.

Die Phasen werden anhand der Schrittmerker ermittelt:

Sprachlich	Anweisungsliste	
	BFZ-SPS (DIN-Variante)	BFZ-SPS (laut FTÜ)
Wenn M00 oder M03 oder M04 oder M05 oder M06 oder M07 oder M10, dann A00	UM00 OM03 OM04 OM05 OM06 OM07 OM10 =A00	*M00 +M03 +M04 +M05 +M06 +M07 +M10 =A00

## Funktionsplan



---

**Aufgabe A6 (SPS)**  
**(Blatt 2 von 3)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

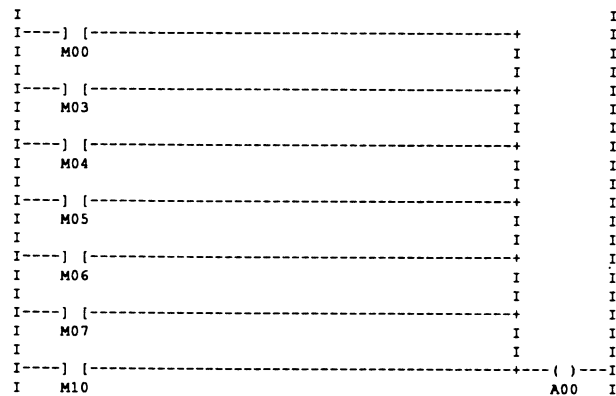
---

**A6d:** Füllen Sie die Leerstellen in dem Ausgabeteil des folgenden SPS-Programms aus. Orientieren Sie sich dazu an dem Signal-Zeit-Diagramm der Aufgabe A6a. Die Zuordnung der logischen Verknüpfungen zu den Phasen entnehmen Sie den Kommentaren.

```
; AUFGABE A6D
; STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
; INITIALISIEREN DER ZEITGEBER
UN M 00      ; ANFANGSBEDINGUNG:
UN M 01      ; ALLE MERKER NICHT
UN M 02      ; GESETZT.
UN M 03      ;
UN M 04      ;
UN M 05      ;
UN M 06      ;
UN M 07      ;
UN M 10      ;
=L Z 00,30    ;1. ROT-GELB-PHASE AMPEL 1
=L Z 01,240   ;2. GRUEN-PHASE      AMPEL 1
=L Z 02,30    ;3. GELB-PHASE      AMPEL 1
=L Z 03,30    ;4. ROT-GELB-PHASE AMPEL 3
=L Z 04,240   ;5. GRUEN-PHASE      AMPEL 3
=L Z 05,30    ;6. GELB-PHASE      AMPEL 3
=L Z 06,30    ;7. PAUSENPHASE      AMPEL 5
=L Z 07,160   ;8. GRUENPHASE      AMPEL 5
=L Z 10,30    ;9. PAUSENPHASE      AMPEL 5
=S A 07      ; FUSSGAENGERAMPELN EINSCHALTEN
=S M 00      ;
=S Z 00      ;1  STARTE PHASE 1
; ABLAUFKETTE (GESTEUERT DURCH ZEITGEBER)
U Z 00      ; PHASE 1 ABGELAUFEN
=R Z 00      ;
=R M 00      ;
=S Z 01      ; STARTE PHASE 2
=S M 01      ;2
U Z 01      ; PHASE 2 ABGELAUFEN
=R Z 01      ;
=R M 01      ;
=S Z 02      ; STARTE PHASE 3
=S M 02      ;3
U Z 02      ; PHASE 3 ABGELAUFEN
=R Z 02      ;
=R M 02      ;
=S Z 03      ; STARTE PHASE 4
=S M 03      ;4
U Z 03      ; PHASE 4 ABGELAUFEN
=R Z 03      ;
=R M 03      ;
=S Z 04      ; STARTE PHASE 5
=S M 04      ;5
U Z 04      ; PHASE 5 ABGELAUFEN
=R Z 04      ;
=R M 04      ;
=S Z 05      ; STARTE PHASE 6
=S M 05      ;6
```



## Kontaktplan



### Hinweise:

Wenn der Kursteilnehmer die Dokumentationsfunktion des MAT 32K benutzt, so kann das abgedruckte Programm unverändert im BFZ-Editor eingegeben und nach Belieben mit Kommentaren versehen werden.

Wenn der Kursteilnehmer den BFZ-SPS-Interpreter in der DIN-Variante benutzen, werden alle Kommentare und alle Leerzeichen in den Anweisungen weggelassen.

Wenn der Kursteilnehmer den BFZ-SPS-Interpreter in der Version der FTÜ benutzt, so wird von nachstehendem gleichwertigen Programm ausgegangen.

```

*/M00*/M01*/M02*/M03*/M04*/M05*/M06*/M07*/M10=LZ00,30=LZ01,240
=LZ02,30=LZ03,30=LZ04,240=LZ05,30=LZ06,30=LZ07,160=LZ10,30
=SA07=SM00=SZ00

```

```
*Z00=RZ00=RM00=SZ01=SM01
*Z01=RZ01=RM01=SZ02=SM02
*Z02=RZ02=RM02=SZ03=SM03
*Z03=RZ03=RM03=SZ04=SM04
*Z04=RZ04=RM04=SZ05=SM05
*Z05=RZ05=RM05=SZ06=SM06
*Z06=RZ06=RM06=SZ07=SM07
*Z07=RZ07=RM07=SZ10=SM10
*Z10=RZ10=RM10=SZ00=SM00
*M00+M03+M04+M05+M06+M07+M10=A00
.....=A01
....=A02
.....=A03
.....=A04
....=A05
....=A06
```

---

**Aufgabe A6 (SPS)**  
**(Blatt 3 von 3)**

---

Name:

---

Kurs:

---

Datum:

---

```
U   Z 05      ;   PHASE 6 ABGELAUFEN
=R  Z 05      ;
=R  M 05      ;
=S  Z 06      ;   STARTE PHASE 7
=S  M 06      ;7
U   Z 06      ;   PHASE 7 ABGELAUFEN
=R  Z 06      ;
=R  M 06      ;
=S  Z 07      ;   STARTE PHASE 8
=S  M 07      ;8
U   Z 07      ;   PHASE 8 ABGELAUFEN
=R  Z 07      ;
=R  M 07      ;
=S  Z 10      ;   STARTE PHASE 9
=S  M 10      ;9
U   Z 10      ;   PHASE 9 ABGELAUFEN
=R  Z 10      ;
=R  M 10      ;
=S  Z 00      ;   STARTE PHASE 1
=S  M 00      ;10
; AUSGABESTEuerung
U   M 00      ;   IN  PHASE 1
O   M 03      ;   UND PHASE 4
O   M 04      ;   UND PHASE 5
O   M 05      ;   UND PHASE 6
O   M 06      ;   UND PHASE 7
O   M 07      ;   UND PHASE 8
O   M 10      ;   UND PHASE 9
=   A 00      ;11 ROT1 SCHALTEN
.   .  .      ;   IN  PHASE 1
.   .  .      ;   UND PHASE 3
=   A 01      ;12 GELB1 SCHALTEN
.   .  .      ;   IN  PHASE 2
=   A 02      ;13 GRUEN1 SCHALTEN
.   .  .      ;   IN  PHASE 1
.   .  .      ;   UND PHASE 2
.   .  .      ;   UND PHASE 3
.   .  .      ;   UND PHASE 4
.   .  .      ;   UND PHASE 7
.   .  .      ;   UND PHASE 8
.   .  .      ;   UND PHASE 9
=   A 03      ;14 ROT3 SCHALTEN
.   .  .      ;   IN  PHASE 4
.   .  .      ;   UND PHASE 6
=   A 04      ;15 GELB3 SCHALTEN
.   .  .      ;   IN  PHASE 5
=   A 05      ;16 GRUEN3 SCHALTEN
.   .  .      ;   IN  PHASE 8
=   A 06      ;17 GRUEN5 SCHALTEN
```

Geben Sie das SPS-Programm ein.  
Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.  
Starten Sie das Programm mit dem Kommando GO.  
Kontrollieren Sie den korrekten Ablauf der Ampelphasen.  
Kontrollieren Sie die Schaltzeiten mit Hilfe einer Stoppuhr.

**4.7 Aufgabe 7: Bedarfssteuerung der Nebenstraße**

Es sollen die Verkehrsampeln 1 und 3 gesteuert werden. Die Fußgängerampeln bleiben ausgeschaltet. Der Verkehr der Nebenstraße unterliegt einer Bedarfssteuerung. Bei hohem Verkehrsaufkommen auf der Nebenstraße darf dennoch eine Minstdauer der Grün-Phase der Hauptstraße nicht unterschritten werden.

Als Schaltzeiten sind festgelegt:

Minstdauer der Grün-Phase der Verkehrsampel 1:	24 Sekunden,
Dauer der Grün-Phase der Verkehrsampel 3:	24 Sekunden,
Dauer der Gelb-Phasen:	3 Sekunden,
Dauer der Rot-Gelb-Phasen:	3 Sekunden.

Bei der Lösung der Aufgabe wird von dem Programm der Aufgabe A5c ausgegangen. Nach dem Ablauf der eigentlichen Grün-Phase der Verkehrsampel 1, die eine Minstdauer für die Freigabe des Verkehrs der Hauptstraße garantieren soll, folgt eine weitere Grün-Phase der gleichen Verkehrsampel. Diese zusätzliche Grün-Phase wird beendet, wenn die "Induktionsschleife" S2 gedrückt wird, also Verkehrsaufkommen von der Nebenstraße vorliegt.

Programmtechnisch erweist sich die zusätzliche Grün-Phase als zusätzlicher Phasenschritt. Er unterscheidet sich von den anderen Phasen durch seine Endbedingung. Während die übrigen Phasen nach Ablauf einer vorgegebenen Zeit enden, endet die zusätzliche Grün-Phase mit dem Eintreffen eines bestimmten Ereignisses des Steuerungsprozesses - hier der Betätigung von S2.

Entsprechend werden die ersten Schritte auch als "zeitgeführte" Schritte und der zusätzliche Schritt als "prozeßgeführter" Schritt bezeichnet.

In der Praxis kommen, wie auch im vorliegenden Beispiel, beide Arten von Schritten meist gemischt vor.

**Hinweis:**

Fortsetzung der Aufgabe A7

- für Assembler-Programmierung auf S. 82,
- für BASIC-Programmierung auf S. 86,
- für SPS-Programmierung auf S. 90.

## Aufgabe A7 (Einführung)

Name: \_\_\_\_\_

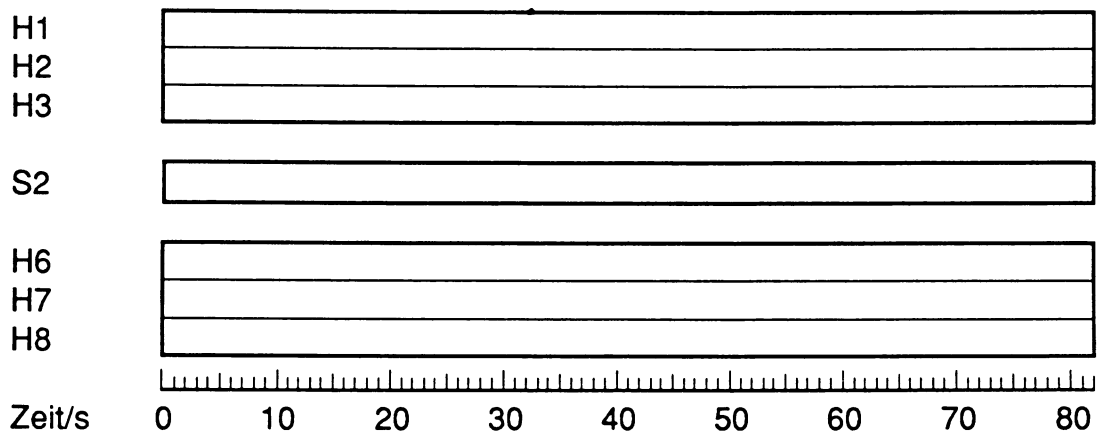
Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

**A7a:** Füllen Sie folgendes Signal-Zeit-Diagramm aus. Schraffieren Sie (vorzugsweise mit Farbstiften) die Zeitabschnitte, in denen die Lampen leuchten. Beginnen Sie mit der Rot-Gelb-Phase der Verkehrsampel 1. Nehmen Sie an, daß zehn Sekunden nach Ablauf der zeitgeführten Grün-Phase der Verkehrsampel 1 die "Induktionsschleife" S2 betätigt wird.

Schreiben Sie die Datenworte, die zu den jeweiligen Abschnitten gehören, unter das Signal-Zeit-Diagramm. Beachten Sie, daß die Fußgängerampeln dunkel bleiben sollen.

Schreiben Sie auch die Datenwörter auf, die von der 8-Bit-Parallel-Eingabe eingelesen werden: vor, nach und während der Betätigung von S2. Gehen Sie dabei davon aus, daß keine anderen Melde-Einrichtungen betätigt werden und die Schalter von E03 bis E07 Low-Pegel erzeugen.

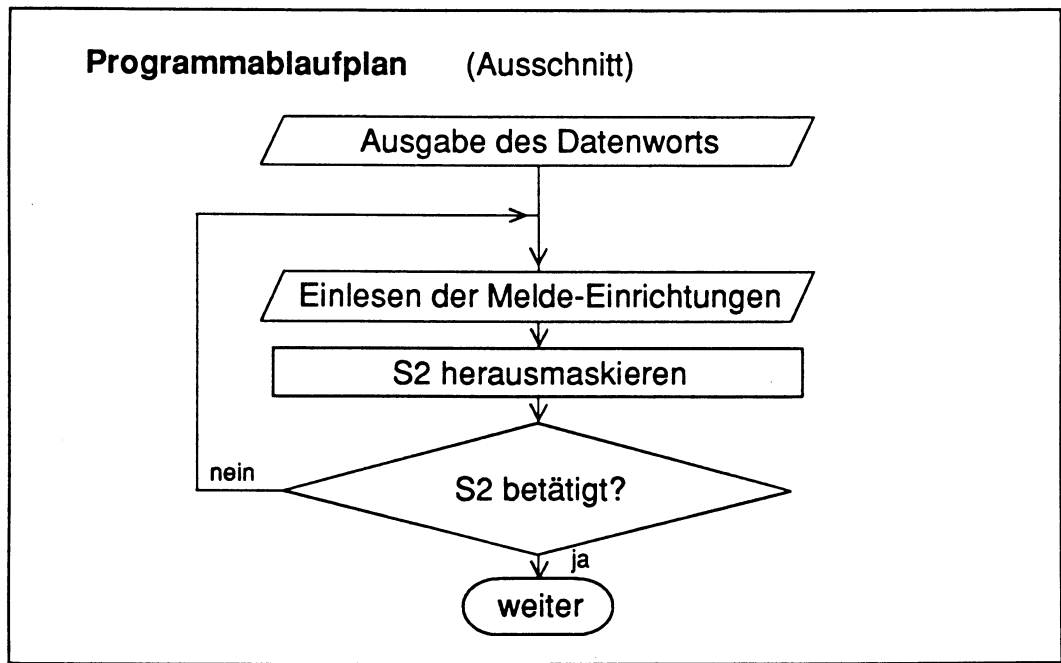


Datenw.  
(Ausgabe)

Datenw.  
(Eingabe)

## 4.7.1 Steuerungsprogramm für die Bedarfssteuerung (Assembler)

Die Programmierung der zeitgeführten Schritte bleibt gegenüber der Lösung der Aufgabe A5c unverändert. Nach dem zweiten Schritt, der zeitgeführt die Minstdauer der Grün-Phase der Verkehrsampel 1 garantiert, wird der zusätzliche prozeßgeführte Schritt eingeführt



In dem vorliegenden Beispiel kann die Ausgabe des Datenworts entfallen, denn es handelt sich hier um das gleiche Datenwort wie im vorangegangenen Schritt (Grün-Phase der Verkehrsampel 1).

---

**Aufgabe A7 (Assembler)**  
**(Blatt 1 von 2)**Name: 

---

Kurs: 

---

Datum: 

---

Das nachstehende Assemblerprogramm steuert die Verkehrsampeln abhängig vom Verkehrsaufkommen der Nebenstraße.

```
0000          ; AUFGABE A7B
0000          ; STEUERUNG DES FAHRZEUGVERKEHRS
0000          ; BEDARFSSTEUERUNG DER SIGNALANLAGE 3
0000          ;
0000          EINGABE EQU    0          ;8-BIT-PARALLEL-EINGABE
0000          AUSGABE EQU    0          ;8-BIT-PARALLEL-AUSGABE
0000          ;
0000          ORG      0E000    ;STARTADRESSE
E000          ;
E000 01 1E00    PHASE1: LXI      B,001E    ;3 SEKUNDEN
E003 3E 0B      MVI      A,0B      ;ROT1 + GELB1 + ROT3
E005 D3 00      OUT      AUSGABE
E007 CD 46E0    CALL     DELAY
E00A 01 F000    PHASE2: LXI      B,00F0    ;24 SEKUNDEN
E00D 3E 0C      MVI      A,0C      ;GRUEN1 + ROT3
E00F D3 00      OUT      AUSGABE
E011 CD 46E0    CALL     DELAY
E014 DB 00      PHAS2A: IN       EINGABE
E016 E6 02      ANI      02          ;TASTE 3 HERAUSMASKIEREN
E018 CA 14E0    JZ       PHAS2A    ;WEITER WARTEN
E01B 01 1E00    PHASE3: LXI      B,001E    ;3 SEKUNDEN
E01E 3E 0A      MVI      A,0A      ;GELB1 + ROT3
E020 D3 00      OUT      AUSGABE
E022 CD 46E0    CALL     DELAY
E025 01 1E00    PHASE4: LXI      B,001E    ;3 SEKUNDEN
E028 3E 19      MVI      A,19      ;ROT1 + ROT3 + GELB3
E02A D3 00      OUT      AUSGABE
E02C CD 46E0    CALL     DELAY
E02F 01 F000    PHASE5: LXI      B,00F0    ;24 SEKUNDEN
E032 3E 21      MVI      A,21      ;ROT1 + GRUEN3
E034 D3 00      OUT      AUSGABE
E036 CD 46E0    CALL     DELAY
E039 01 1E00    PHASE6: LXI      B,001E    ;3 SEKUNDEN
E03C 3E 11      MVI      A,11      ;ROT1 + GELB3
E03E D3 00      OUT      AUSGABE
E040 CD 46E0    CALL     DELAY
E043          ;
E043 C3 00E0    JMP      PHASE1    ;ENDLOSSCHLEIFE
E046          ;
E046          ;      UNTERPROGRAMM DELAY
E046 C5        DELAY: PUSH     B          ;REGISTER RETTEN
E047 D5        PUSH     D
E048 F5        PUSH     PSW
E049 11 3425    DELAY1: LXI      D,2534    ;ZEITBASIS 1/10 S
E04C 1B        DELAY2: DCX      D          ;GEGEN NULL ZAEHLEN
E04D 7A        MOV      A,D          ;TESTE AUF NULL
E04E B3        ORA      E
```

Die neue, zusätzliche Phase (prozeßgeführter Schritt) wird folgendermaßen programmiert:

PHAS2A: IN 0	Das Datenwort der 8-Bit-Parallel-Eingabe (Adresse 0) wird in den Akkumulator eingelesen.
ANI 02	Der Inhalt des Akkumulators wird einer UND-Verknüpfung mit der Konstanten 02 unterzogen, um isoliert das Bit des Eingangs E01 zu untersuchen.
JZ PHAS2A	Wenn der Inhalt des Akkumulators null ist (S2 nicht betätigt), wird zu der Marke PHAS2A verzweigt. Damit werden die Befehle dieses Programmschritts wiederholt, der prozeßgeführte Schritt dauert an. Wenn der Inhalt des Akkumulators von null verschieden ist (S2 betätigt), wird in der nächsten Zeile fortgefahren. Der prozeßgeführte Schritt ist beendet.

**Hinweis:**

Das Programm kann auch von der Diskette geladen werden (Datei A7B.ASM bzw. Datei A7B.MAT), da es vom Kursteilnehmer nicht vervollständigt werden muß. Damit wird Zeit gewonnen, um die beschriebenen Tests sorgfältig durchführen zu können.

---

**Aufgabe A7 (Assembler)**  
**(Blatt 2 von 2)**

---

Name:

---

Kurs:

---

Datum:

---

E04F C2 4CE0	JNZ	DELAY2	;SCHLEIFENENDE DELAY2
E052 0B	DCX	B	;GEGEN NULL ZAEHLEN
E053 78	MOV	A,B	;TESTE AUF NULL
E054 B1	ORA	C	
E055 C2 49E0	JNZ	DELAY1	;SCHLEIFENENDE DELAY1
E058 F1	POP	PSW	
E059 D1	POP	D	
E05A C1	POP	B	
E05B C9	RET		;UNTERPROGRAMM DELAY

Geben Sie das Programm mit Hilfe des Memory- oder Assembler-Kommandos des MAT 85 ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des Print- oder Disassembler-Kommandos.

Starten Sie das Programm mit dem GO-Kommando.

Beantworten Sie die folgenden Fragen. Versuchen Sie jeweils zunächst, die Antwort aus dem Verständnis des Programms zu geben. Überprüfen Sie anschließend Ihre Antwort im Experiment.

**A7b:** Das Programm ist gestartet, die Verkehrsampel 1 zeigt grün an. Warten Sie ggf. noch 24 Sekunden. Drücken Sie dann kurz die Taste S2. Was geschieht?

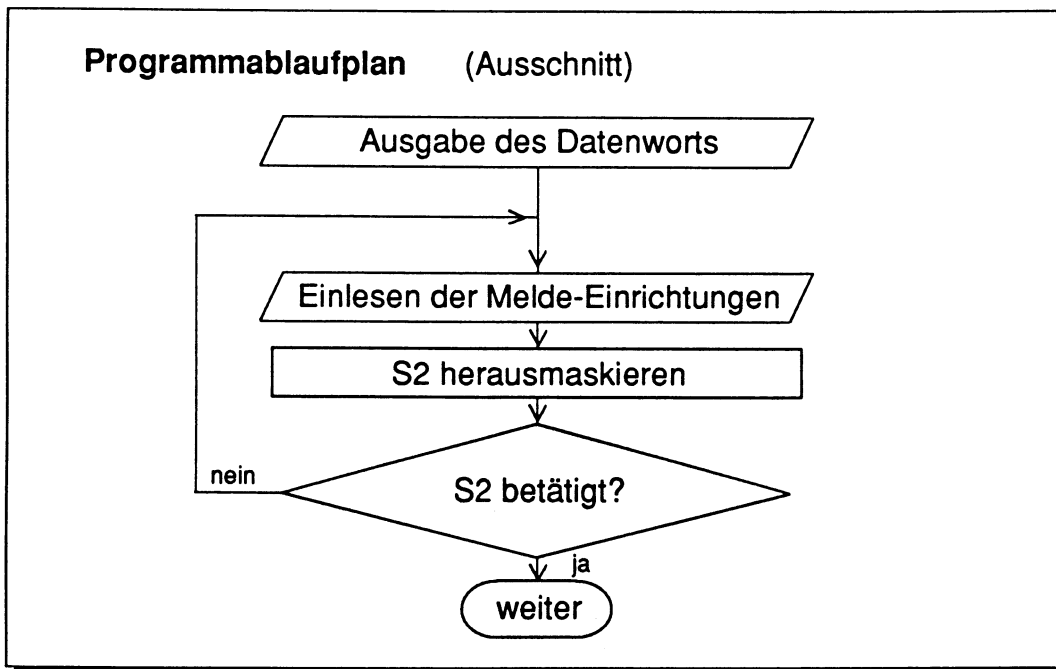
**A7c:** Drücken Sie so lange die Taste S2, bis die Gelb-Phase der Verkehrsampel 1 erscheint. Lassen Sie S2 los. Drücken Sie S2 erneut während der Gelb-Phase der Verkehrsampel 3 und halten Sie S2 fest. Wie lange dauert die Grün-Phase der Verkehrsampel 1?

**A7d:** Starten Sie das Programm erneut. Gleich nach Erscheinen der Grün-Phase der Verkehrsampel 1 drücken Sie kurz S2. Was geschieht?



## 4.7.2 Steuerungsprogramm für die Bedarfssteuerung (BASIC)

Die Programmierung der zeitgeführten Schritte bleibt gegenüber der Lösung der Aufgabe A5c unverändert. Nach dem zweiten Schritt, der zeitgeführt die Minstdauer der Grün-Phase der Verkehrsampel 1 garantiert, wird der zusätzliche prozeßgeführte Schritt eingeführt. Er wird durch folgenden Programmablaufplan beschrieben:



In dem vorliegenden Beispiel kann die Ausgabe des Datenworts entfallen, denn es handelt sich hier um das gleiche Datenwort wie im vorangegangenen Schritt (Grün-Phase der Verkehrsampel 1).

---

**Aufgabe A7 (BASIC)**  
**(Blatt 1 von 2)**

---

Name: 

---

---

Kurs: 

---

---

Datum: 

---

Das nachstehende Assemblerprogramm steuert die Verkehrsampeln abhängig vom Verkehrsaufkommen der Nebenstraße.

```
10 REM AUFGABE A7B
20 REM STEUERUNG DES FAHRZEUGVERKEHRS
30 REM BEDARFSSTEUERUNG DER SIGNALANLAGE 3
40 REM
50 A=0                      :REM 8-BIT-PARALLEL-AUSGABE
60 E=0                      :REM 8-BIT-PARALLEL-EINGABE
100 Z=30                    :REM 3 SEKUNDEN
110 OUT A,11                :REM ROT1 + GELB1 + ROT3
120 GOSUB 1000
200 Z=240                   :REM 24 SEKUNDEN
210 OUT A,12                :REM GRUEN1 + ROT3
220 GOSUB 1000
230 T=INP(E)                :REM TASTEN EINLESEN
240 IF (T AND 2) = 0 THEN GOTO 230
300 Z=30                    :REM 3 SEKUNDEN
310 OUT A,10                :REM GELB1 + ROT3
320 GOSUB 1000
400 Z=30                    :REM 3 SEKUNDEN
410 OUT A,25                :REM ROT1 + ROT3 + GELB3
420 GOSUB 1000
500 Z=240                   :REM 24 SEKUNDEN
510 OUT A,33                :REM ROT1 + GRUEN3
520 GOSUB 1000
600 Z=30                    :REM 3 SEKUNDEN
610 OUT A,17                :REM ROT1 + GELB3
620 GOSUB 1000
999 GOTO 100
1000 REM VERZOEGERUNG (Z = ZEIT IN 1/10 SEKUNDEN)
1010 FOR I=1 TO Z
1020 FOR J=1 TO 98
1030 NEXT J
1040 NEXT I
1050 RETURN
```

Geben Sie das Programm ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.

Starten Sie das Programm mit dem Kommando RUN.

Beantworten Sie die folgenden Fragen. Versuchen Sie jeweils zunächst, die Antwort aus dem Verständnis des Programms zu geben. Überprüfen Sie anschließend Ihre Antwort im Experiment.

Die neue, zusätzliche Phase (prozeßgeführter Schritt) wird folgendermaßen programmiert:

230	T=INP (0)	Das Datenwort der 8-Bit-Parallel-Eingabe (Adresse 0) wird in die Variable T eingelesen.
240	T=T AND 2	Der Inhalt der Variablen T wird einer UND-Verknüpfung mit der Konstanten 2 unterzogen, um isoliert das Bit des Eingangs E01 zu untersuchen.
250	IF T=0 THEN GOTO 230	Wenn der Inhalt des Akkumulators null ist (S2 nicht betätigt), wird zu der Zeile 230 verzweigt. Damit werden die Befehle dieses Programmschritts wiederholt, der prozeßgeführte Schritt dauert an.
300	...	Wenn der Inhalt des Akkumulators von null verschieden ist (S2 betätigt), wird in der nächsten Zeile fortgefahren. Der prozeßgeführte Schritt ist beendet.

**Hinweis:**

In dem Programm der Aufgabenstellung wurden die hier aufgeführten Zeilen 240 und 250 zusammengefaßt.

**Hinweis:**

Das Programm kann auch von der Diskette geladen werden (Datei A7B.BAS bzw. Datei A7B.B6K), da es vom Kursteilnehmer nicht vervollständigt werden muß. Damit wird Zeit gewonnen, um die beschriebenen Tests sorgfältig durchführen zu können.

---

**Aufgabe A7 (BASIC)**  
**(Blatt 2 von 2)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

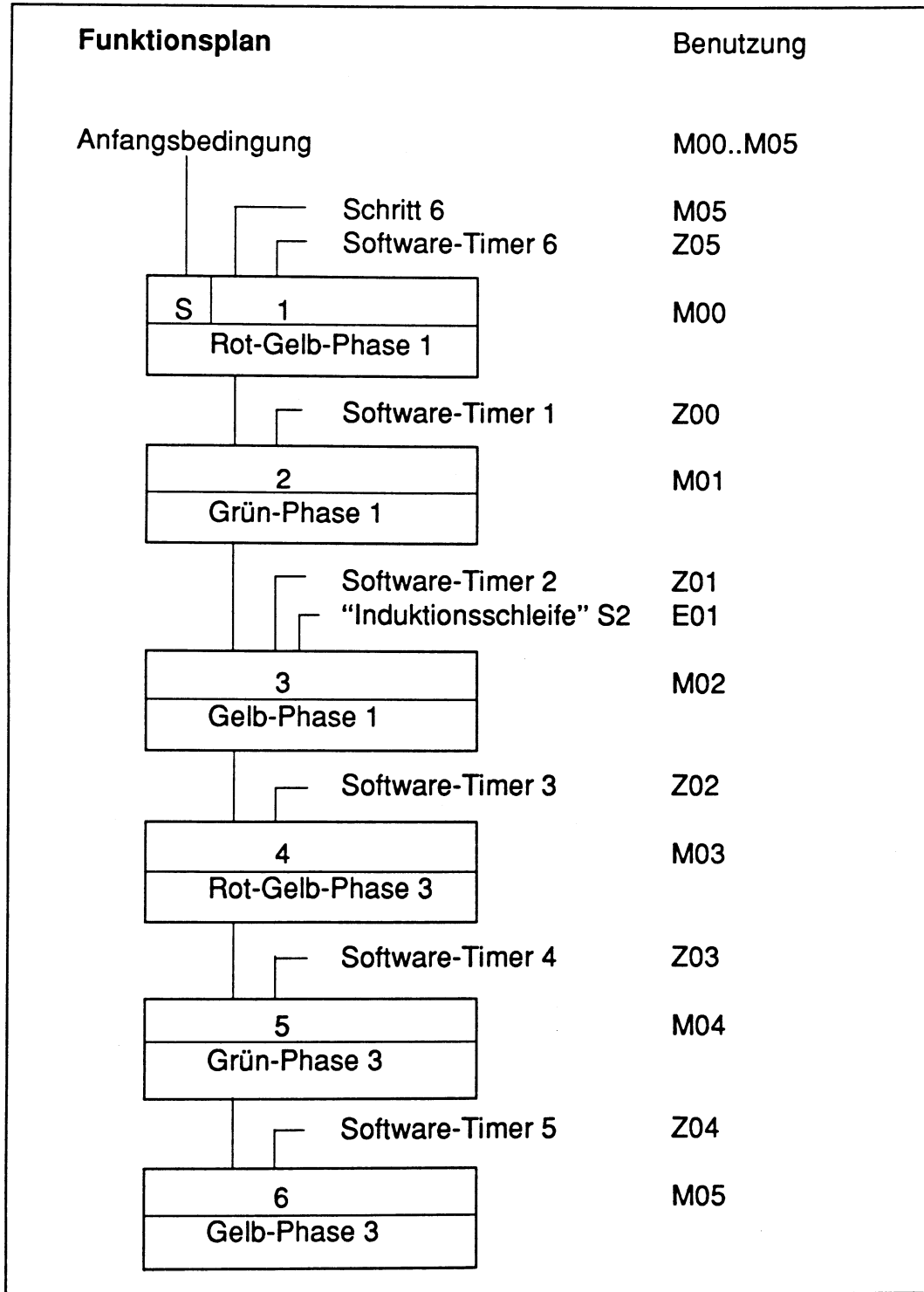
**A7b:** Das Programm ist gestartet, die Verkehrsampel 1 zeigt grün an. Warten Sie ggf. noch 24 Sekunden. Drücken Sie dann kurz die Taste S2. Was geschieht?

**A7c:** Drücken Sie so lange die Taste S2, bis die Gelb-Phase der Verkehrsampel 1 erscheint. Lassen Sie S2 los. Drücken Sie S2 erneut während der Gelb-Phase der Verkehrsampel 3 und halten Sie S2 fest. Wie lange dauert die Grün-Phase der Verkehrsampel 1?

**A7d:** Starten Sie das Programm erneut. Gleich nach Erscheinen der Grün-Phase der Verkehrsampel 1 drücken Sie kurz S2. Was geschieht?

## 4.7.3 Steuerungsprogramm für die Bedarfssteuerung (SPS)

Die Programmierung der zeitgeführten Schritte bleibt gegenüber der Aufgabe A5c weitgehend unverändert.



---

**Aufgabe A7 (SPS)**  
**(Blatt 1 von 2)**

---

Name: 

---

---

Kurs: 

---

---

Datum: 

---

Das nachstehende Assemblerprogramm steuert die Verkehrsampeln abhängig vom Verkehrsaufkommen der Nebenstraße.

```
; AUFGABE A7B
; STEUERUNG DES FAHRZEUGVERKEHRS, BEDARFSSTEUERUNG DER NEBENSTR.
; INITIALISIEREN DER ZEITGEBER
UN M 00      ; ANFANGSBEDINGUNG:
UN M 01      ; ALLE MERKER NICHT
UN M 02      ; GESETZT.
UN M 03      ;
UN M 04      ;
UN M 05      ;
=L Z 00,30    ;1. ROT-GELB-PHASE AMPEL 1
=L Z 01,240   ;2. GRUEN-PHASE      AMPEL 1
=L Z 02,30    ;3. GELB-PHASE      AMPEL 1
=L Z 03,30    ;4. ROT-GELB-PHASE AMPEL 3
=L Z 04,240   ;5. GRUEN-PHASE      AMPEL 3
=L Z 05,30    ;6. GELB-PHASE      AMPEL 3
=S M 00      ;
=S Z 00      ;1  STARTE PHASE 1
; ABLAUFKETTE (GESTEUERT DURCH ZEITGEBER)
U Z 00      ; PHASE 1 ABGELAUFEN
=R Z 00      ;
=R M 00      ;
=S Z 01      ; STARTE PHASE 2
=S M 01      ;2
U Z 01      ; PHASE 2 ABGELAUFEN
U E 01      ; UND FAHRZEUG AN IND.SHL. 3
=R Z 01      ;
=R M 01      ;
=S Z 02      ; STARTE PHASE 3
=S M 02      ;3
U Z 02      ; PHASE 3 ABGELAUFEN
=R Z 02      ;
=R M 02      ;
=S Z 03      ; STARTE PHASE 4
=S M 03      ;4
U Z 03      ; PHASE 4 ABGELAUFEN
=R Z 03      ;
=R M 03      ;
=S Z 04      ; STARTE PHASE 5
=S M 04      ;5
U Z 04      ; PHASE 5 ABGELAUFEN
=R Z 04      ;
=R M 04      ;
=S Z 05      ; STARTE PHASE 6
=S M 05      ;6
U Z 05      ; PHASE 6 ABGELAUFEN
=R Z 05      ;
=R M 05      ;
=S Z 00      ; STARTE PHASE 1
=S M 00      ;7
```

Lediglich der Übergang vom zweiten zum dritten Schritt wurde neu gestaltet. Er enthält zwei Bedingungen, die beide erfüllt sein müssen:

- der Software-Timer des Schritts 2 muß abgelaufen sein,
- der Eingang E01 muß gesetzt sein.

Damit ist gewährleistet, daß die Grün-Phase der Verkehrsampel 3 auch bei hohem Verkehrsaufkommen der Nebenstraße für eine Mindestdauer geschaltet wird. Zum andern enthält der Übergang die Bedarfssteuerung der Nebenstraße. Erst wenn S2 betätigt wird, wird der Übergang durchgeführt, d.h. schaltet Verkehrsampel 3 auf gelb. Die weiteren Phasen laufen zeitgesteuert ab, so daß die Nebenstraße zum Zuge kommt und schließlich die Steuerung wieder in der Grün-Phase der Hauptstraße verharret.

Durch die Einführung der zusätzlichen Übergangsbedingung ist für den "prozeßgeführten" Teil der Grün-Phase der Verkehrsampel 3 kein weiterer Schrittmerker notwendig.

**Hinweise:**

Wenn der Kursteilnehmer den BFZ-SPS-Interpreter in der Version der FTÜ benutzt, so wird von nachstehendem gleichwertigen Programm ausgegangen.

```
* /M00*/M01*/M02*/M03*/M04*/M05=LZ00,30=LZ01,240
=LZ02,30=LZ03,30=LZ04,240=LZ05,30=SM00=SZ00
*M00=RZ00=RM00=SZ01=SM01
*M01=E01=RZ01=RM01=SZ02=SM02
*M02=RZ02=RM02=SZ03=SM03
*M03=RZ03=RM03=SZ04=SM04
*M04=RZ04=RM04=SZ05=SM05
*M05=RZ05=RM05=SZ00=SM00
*M00+M03+M04+M05=A00
*M00+M02=A01
*M01=A02
*M00+M01+M02+M03=A03
*M03+M05=A04
*M04=A05
```

Die Programme können auch von der Diskette geladen werden (Dateien A7B.DOK bzw. A7B.DIN bzw. A7B.SPS), da sie vom Kursteilnehmer nicht vervollständigt werden müssen. Damit wird Zeit gewonnen, um die beschriebenen Tests sorgfältig durchzuführen.

---

**Aufgabe A7 (SPS)**  
**(Blatt 2 von 2)**

---

Name: 

---

---

Kurs: 

---

---

Datum: 

---

```
; AUSGABESTEuerung
U  M 00      ; IN  PHASE 1
O  M 03      ; UND PHASE 4
O  M 04      ; UND PHASE 5
O  M 05      ; UND PHASE 6
=  A 00      ;8  ROT1 SCHALTEN
U  M 00      ; IN  PHASE 1
O  M 02      ; UND PHASE 3
=  A 01      ;9  GELB1 SCHALTEN
U  M 01      ; IN  PHASE 2
=  A 02      ;10 GRUEN1 SCHALTEN
U  M 00      ; IN  PHASE 1
O  M 01      ; UND PHASE 2
O  M 02      ; UND PHASE 3
O  M 03      ; UND PHASE 4
=  A 03      ;11 ROT3 SCHALTEN
U  M 03      ; IN  PHASE 4
O  M 05      ; UND PHASE 6
=  A 04      ;12 GELB3 SCHALTEN
U  M 04      ; IN  PHASE 5
=  A 05      ;13 GRUEN3 SCHALTEN
```

Geben Sie das Programm ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.

Starten Sie das Programm mit dem Kommando GO.

Beantworten Sie die folgenden Fragen. Versuchen Sie jeweils zunächst, die Antwort aus dem Verständnis des Programms zu geben. Überprüfen Sie anschließend Ihre Antwort im Experiment.

**A7b:** Das Programm ist gestartet, die Verkehrsampel 1 zeigt grün an. Warten Sie ggf. noch 24 Sekunden. Drücken Sie dann kurz die Taste S2. Was geschieht?

**A7c:** Drücken Sie so lange die Taste S2, bis die Gelb-Phase der Verkehrsampel 1 erscheint. Lassen Sie S2 los. Drücken Sie S2 erneut während der Gelb-Phase der Verkehrsampel 3 und halten Sie S2 fest. Wie lange dauert die Grün-Phase der Verkehrsampel 1?

**A7d:** Starten Sie das Programm erneut. Gleich nach Erscheinen der Grün-Phase der Verkehrsampel 1 drücken Sie kurz S2. Was geschieht?



**4.8 Aufgabe 8: Bedarfssteuerung der Nebenstraße und des Fußwegs**

Es sollen die Verkehrsampeln 1 und 3 sowie die Fußgängerampel 5 gesteuert werden. Der Verkehr der Nebenstraße sowie das Überqueren der Hauptstraße nach der Einmündung unterliegen einer Bedarfssteuerung. Bei hohem Verkehrsaufkommen auf der Nebenstraße und am Fußgängerübergang darf dennoch eine Minstdauer der Grün-Phase der Hauptstraße nicht unterschritten werden.

Als Schaltzeiten sind festgelegt:

Minstdauer der Grün-Phase der Verkehrsampel 1:	24 Sekunden,
Dauer der Grün-Phase der Verkehrsampel 3:	24 Sekunden,
Dauer der Gelb-Phasen:	3 Sekunden,
Dauer der Rot-Gelb-Phasen:	3 Sekunden,
Dauer der Grün-Phase der Fußgängerampel 5:	16 Sekunden,
Dauer der Übergangsphase der Fußgängerampel 5:	3 Sekunden.

Bei der Lösung der Aufgabe wird von dem Programm der Aufgabe A6c ausgegangen. Nach dem Ablauf der eigentlichen Grün-Phase der Verkehrsampel 1, die eine Minstdauer für die Freigabe des Verkehrs der Hauptstraße garantieren soll, folgt eine weitere Grün-Phase der gleichen Verkehrsampel. Diese zusätzliche Grünphase wird beendet, wenn entweder

- a) die "Induktionsschleife" S2 gedrückt wird, also Verkehrsaufkommen von der Nebenstraße vorliegt
- oder
- b) Die Meldetaste S3 an der Fußgängerampel 5 gedrückt wird, also ein Fußgänger die Straße überqueren will.

Programmtechnisch erweist sich die zusätzliche Grünphase als zusätzlicher Phasenschritt wie schon in Aufgabe 7.

**Aufgabe A8 (Einführung)**  
**(Blatt 1 von 3)**

Name: \_\_\_\_\_

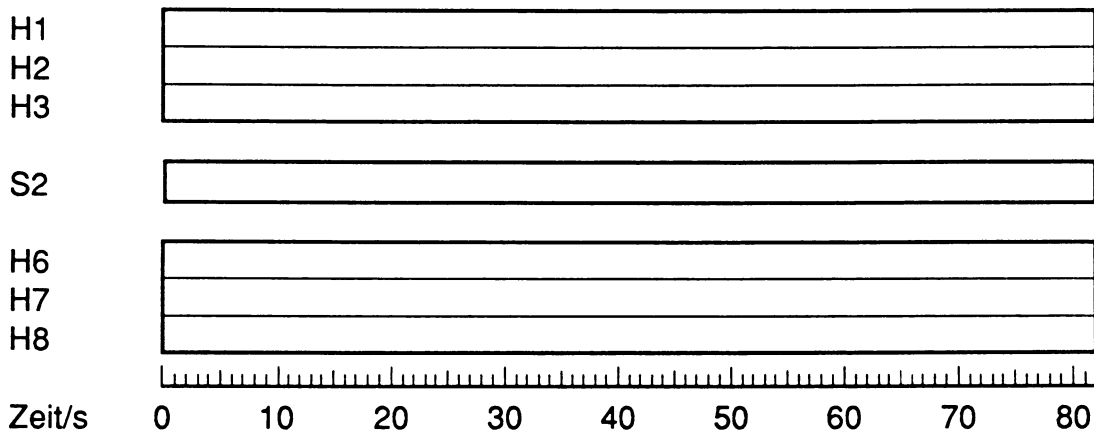
Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

**A8a:** Füllen Sie folgendes Signal-Zeit-Diagramm aus. Schraffieren Sie (vorzugsweise mit Farbstiften) die Zeitabschnitte in denen die Lampen leuchten. Beginnen Sie mit der Rot-Gelb-Phase der Verkehrsampel 1. Nehmen Sie an, daß zehn Sekunden nach Ablauf der zeitgeführten Grün-Phase der Verkehrsampel 1 die "Induktionsschleife" S2 betätigt wird.

Schreiben Sie die Datenwörter, die zu den jeweiligen Abschnitten gehören, unter das Signal-Zeit-Diagramm. Beachten Sie, daß die Fußgängerampeln in Betrieb sind.

Schreiben Sie auch die Datenwörter auf, die von der 8-Bit-Parallel-Eingabe eingelesen werden: vor, nach und während der Betätigung von S2. Gehen Sie dabei davon aus, daß keine anderen Melde-Einrichtungen betätigt werden und die Schalter E03 bis E07 Low-Pegel erzeugen.



Datenw.  
(Ausgabe)

Datenw.  
(Eingabe)

Nach Ablauf der Gelb-Phase der Verkehrsampel 1 muß erneut geprüft werden, weshalb die Grün-Phase beendet wurde. War es die "Induktionsschleife" S2, so müssen anschließend die Rot-Gelb-Phase, die Grün-Phase und die Gelb-Phase der Verkehrsampel 3 durchlaufen werden.

War es die Meldetaste S3, so schaltet das Programm die Übergangsphase 1, die Grün-Phase und die Übergangsphase 2 der Fußgängerampel.

An dieser Stelle tritt eine Programmverzweigung auf. Der weitere Verlauf des Programms hängt von einem vorher ermittelten Ergebnis ab, in diesem Fall vom Einlesen der Eingabeleitungen und der entsprechenden Entscheidung, welche der oben genannten Situationen vorliegt.

**Hinweis:**

Fortsetzung der Aufgabe A8

- für Assembler-Programmierung auf S. 100,
- für BASIC-Programmierung auf S. 106,
- für SPS-Programmierung auf S. 110.

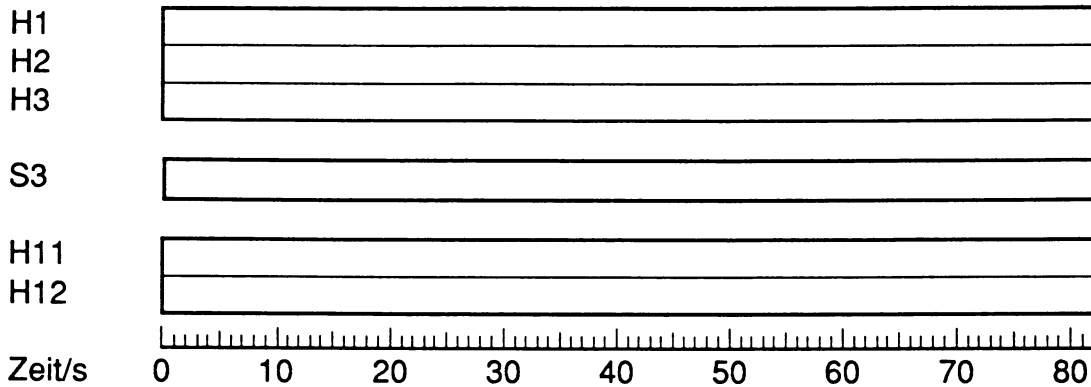
**Aufgabe A8 (Einführung)**  
**(Blatt 2 von 3)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

**A8b:** Füllen Sie wie in Aufgabe A8a folgendes Signal-Zeit-Diagramm aus. Beginnen Sie mit der Rot-Gelb-Phase der Verkehrsampel 1. Nehmen Sie an, daß zehn Sekunden nach Ablauf der zeitgeführten Grün-Phase der Verkehrsampel 1 die Meldetaste S3 betätigt wird.



Datenwörter  
(Ausgabe)

Datenwörter  
(Eingabe)

---

#### 4. Aufgaben

---

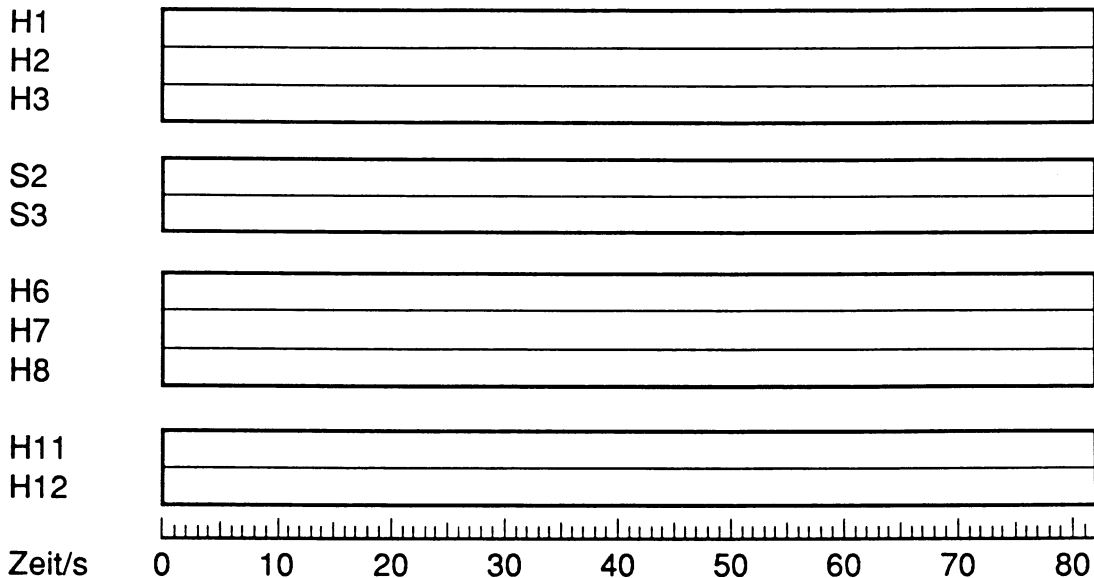
**Aufgabe A8 (Einführung)**  
**(Blatt 3 von 3)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

**A8c:** Füllen Sie wie in Aufgabe A8a folgendes Signal-Zeit-Diagramm aus. Beginnen Sie mit der Rot-Gelb-Phase der Verkehrsampel 1. Nehmen Sie an, daß noch während der zeitgeführten Grün-Phase der Verkehrsampel 1 die "Induktionsschleife" S2 und die Meldetaste S3 betätigt und festgehalten werden. Die Tasten werden jeweils wieder losgelassen, wenn für den entsprechenden Verkehrsweg "Grün" gemeldet wird.

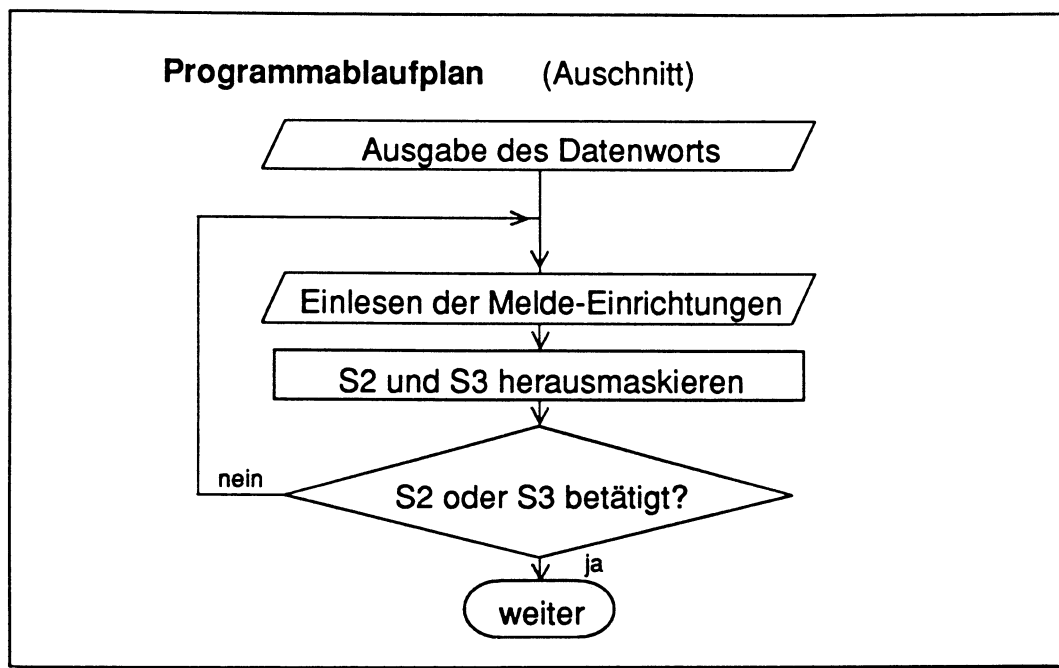


Datenwörter  
(Ausgabe)

Datenwörter  
(Eingabe)

**4.8.1 Erweiterte Bedarfssteuerung (Assembler)**

Die Programmierung der zeitgeführten Schritte bleibt gegenüber der Lösung der Aufgabe A6c unverändert. Nach dem zweiten Schritt, der zeitgeführt die Minstdauer der Grün-Phase der Verkehrsampel 1 garantiert, wird der zusätzliche prozeßgeführte Schritt eingeführt. Er wird durch folgenden Programmablaufplan beschrieben:



In dem vorliegenden Beispiel kann die Ausgabe des Datenworts entfallen, denn es handelt sich hier um das gleiche Datenwort wie im vorangegangenen Schritt (Grün-Phase der Verkehrsampel 1).

---

**Aufgabe A8 (Assembler)**  
**(Blatt 1 von 3)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

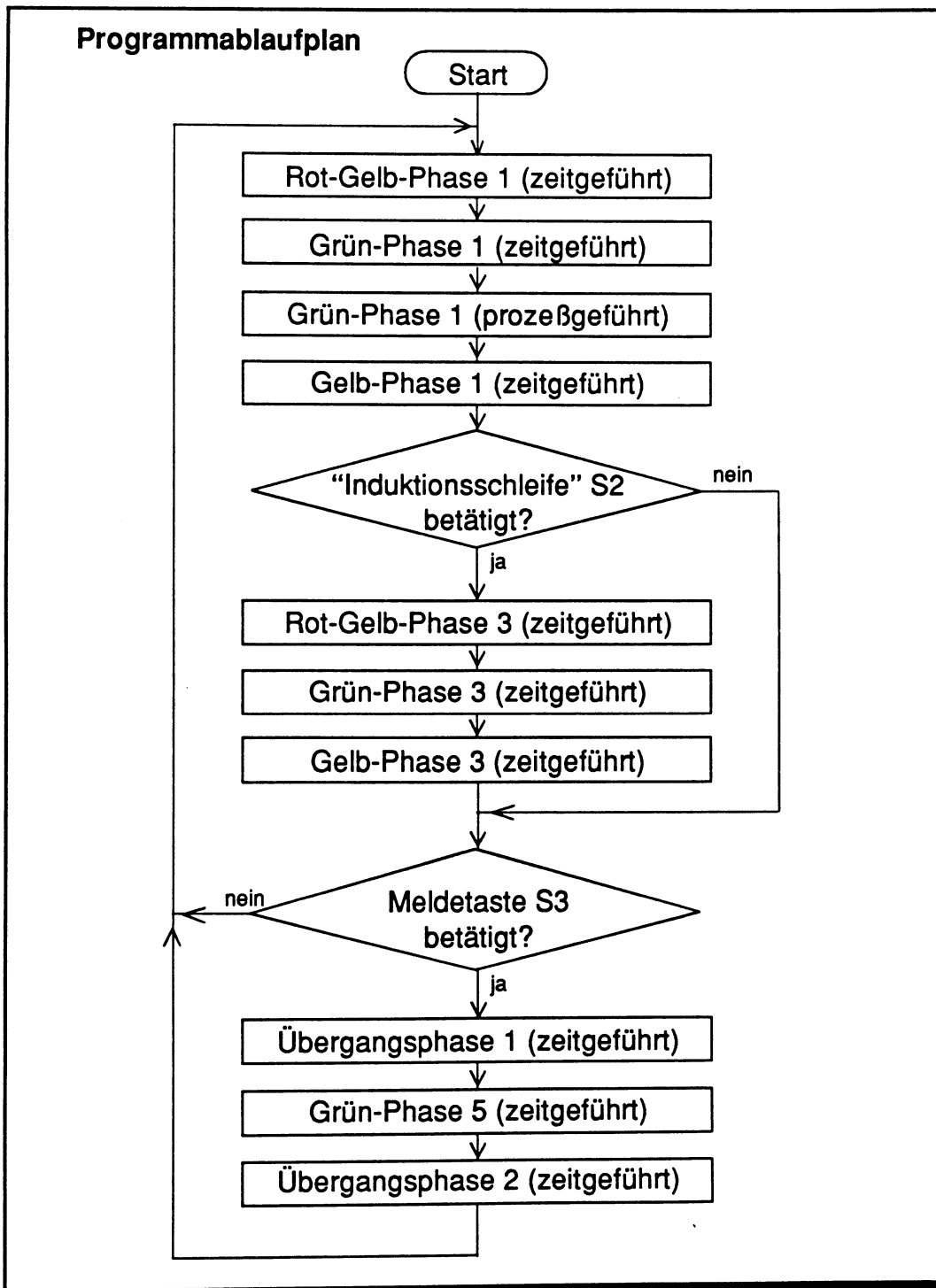
---

Das nachstehende Assemblerprogramm steuert die Ampeln abhängig vom Verkehrsaufkommen der Nebenstraße und vom Fußgängerverkehr.

```
0000          ; AUFGABE A8D
0000          ; STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
0000          ; BEDARFSSTEUERUNG DER SIGNALANLAGEN 3 UND 5
0000          ;
0000  EINGABE EQU    0          ; 8-BIT-PARALLEL-EINGABE
0000  AUSGABE EQU   0          ; 8-BIT-PARALLEL-AUSGABE
0000          ;
0000          ORG      0E000    ; STARTADRESSE
E000          ;
E000 01 1E00    PHASE1: LXI    B,001E ; 3 SEKUNDEN
E003 3E 8B      MVI      A,8B   ; ROT1 + GELB1 + ROT3
E005 D3 00      OUT      AUSGABE
E007 CD 79E0    CALL     DELAY
E00A 01 F000    PHASE2: LXI    B,00F0 ; 24 SEKUNDEN
E00D 3E 8C      MVI      A,8C   ; GRUEN1 + ROT3
E00F D3 00      OUT      AUSGABE
E011 CD 79E0    CALL     DELAY
E014 DB 00      PHAS2A: IN     EINGABE ; LIES TASTENSTELLUNGEN
E016 E6 06      ANI      06     ; TASTE 3 UND TASTE 5 HERAUSMASKIEREN
E018 32 8FE0    STA      TASTEN ; ABSPEICHERN
E01B CA 14E0    JZ       PHAS2A ; WEITER WARTEN
E01E 01 1E00    PHASE3: LXI    B,001E ; 3 SEKUNDEN
E021 3E 8A      MVI      A,8A   ; GELB1 + ROT3
E023 D3 00      OUT      AUSGABE
E025 CD 79E0    CALL     DELAY
E028 3A 8FE0    PHASE4: LDA     TASTEN ; TASTENSTELLUNGEN
E02B B7         ORA      A      ; FLAGGEN SETZEN
E02C E6 02      ANI      02     ; TASTE 3 GEDRUECKT?
E02E CA 4FE0    JZ       PHASE7 ; NEIN - WEITER
E031 01 1E00    LXI      B,001E ; 3 SEKUNDEN
E034 3E 99      MVI      A,99   ; ROT1 + ROT3 + GELB3
E036 D3 00      OUT      AUSGABE
E038 CD 79E0    CALL     DELAY
E03B 01 F000    PHASE5: LXI    B,00F0 ; 24 SEKUNDEN
E03E 3E A1      MVI      A,0A1   ; ROT1 + GRUEN3
E040 D3 00      OUT      AUSGABE
E042 CD 79E0    CALL     DELAY
E045 01 1E00    PHASE6: LXI    B,001E ; 3 SEKUNDEN
E048 3E 91      MVI      A,91   ; ROT1 + GELB3
E04A D3 00      OUT      AUSGABE
E04C CD 79E0    CALL     DELAY
E04F 3A 8FE0    PHASE7: LDA     TASTEN ; TASTENSTELLUNGEN
E052 B7         ORA      A      ; FLAGGEN SETZEN
E053 E6 04      ANI      04     ; TASTE 5 GEDRUECKT?
E055 CA 00E0    JZ       PHASE1 ; NEIN - WEITER
E058 01 1E00    LXI      B,001E ; 3 SEKUNDEN
```



Die Programmverzweigung wird nachfolgend in einem Programmablaufplan dargestellt. Dieser dient der strukturellen Übersicht über das Gesamtprogramm und löst die Programmabschnitte nicht bis in die einzelnen Schritte auf.



---

**Aufgabe A8 (Assembler)**  
**(Blatt 2 von 3)**

---

Name:

---

Kurs:

---

Datum:

---

```
E05B 3E 89          MVI    A,89      ;ROT1 + ROT3
E05D D3 00          OUT     AUSGABE
E05F CD 79E0        CALL    DELAY
E062 01 A000    PHASE8: LXI    B,00A0  ;16 SEKUNDEN
E065 3E C9          MVI    A,0C9     ;ROT1 + ROT3 + GRUEN5
E067 D3 00          OUT     AUSGABE
E069 CD 79E0        CALL    DELAY
E06C 01 1E00    PHASE9: LXI    B,001E  ;3 SEKUNDEN
E06F 3E 89          MVI    A,89      ;ROT1 + ROT3
E071 D3 00          OUT     AUSGABE
E073 CD 79E0        CALL    DELAY
E076                ;
E076 C3 00E0        JMP     PHASE1    ;ENDLOSSCHLEIFE
E079                ;
E079                ;    UNTERPROGRAMM DELAY
E079 C5            DELAY: PUSH    B      ;REGISTER RETTEN
E07A D5            PUSH    D
E07B F5            PUSH    PSW
E07C 11 3425    DELAY1: LXI    D,2534  ;ZEITBASIS 1/10 S
E07F 1B            DELAY2: DCX    D      ;GEGEN NULL ZAEHLEN
E080 7A            MOV     A,D      ;TESTE AUF NULL
E081 B3            ORA     E
E082 C2 7FE0        JNZ     DELAY2    ;SCHLEIFENENDE DELAY2
E085 0B            DCX     B      ;GEGEN NULL ZAEHLEN
E086 78            MOV     A,B      ;TESTE AUF NULL
E087 B1            ORA     C
E088 C2 7CE0        JNZ     DELAY1    ;SCHLEIFENENDE DELAY1
E08B F1            POP     PSW
E08C D1            POP     D
E08D C1            POP     B
E08E C9            RET              ;UNTERPROGRAMM DELAY
E08F                ;
E08F                ; DATENSPEICHER
E08F 00            TASTEN: DB      0      ;SPEICHER FUER EINGABEBYTE
```

Geben Sie das Programm mit Hilfe des Memory- oder Assembler-Kommandos des MAT 85 ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des Print- oder Disassembler-Kommandos.

Starten Sie das Programm mit dem GO-Kommando.

**Hinweis:**

Das Programm kann auch von der Diskette geladen werden (Datei A8D.ASM bzw. Datei A8D.MAT), da es vom Kursteilnehmer nicht vervollständigt werden muß. Damit wird Zeit gewonnen, um die nachfolgend beschriebenen Tests sorgfältig durchzuführen.

---

**Aufgabe A8 (Assembler)**  
**(Blatt 3 von 3)**

---

Name: 

---

---

Kurs: 

---

---

Datum: 

---

Beantworten Sie die folgenden Fragen. Versuchen Sie jeweils zunächst, die Antwort aus dem Verständnis des Programms zu geben. Überprüfen Sie anschließend Ihre Antwort im Experiment.

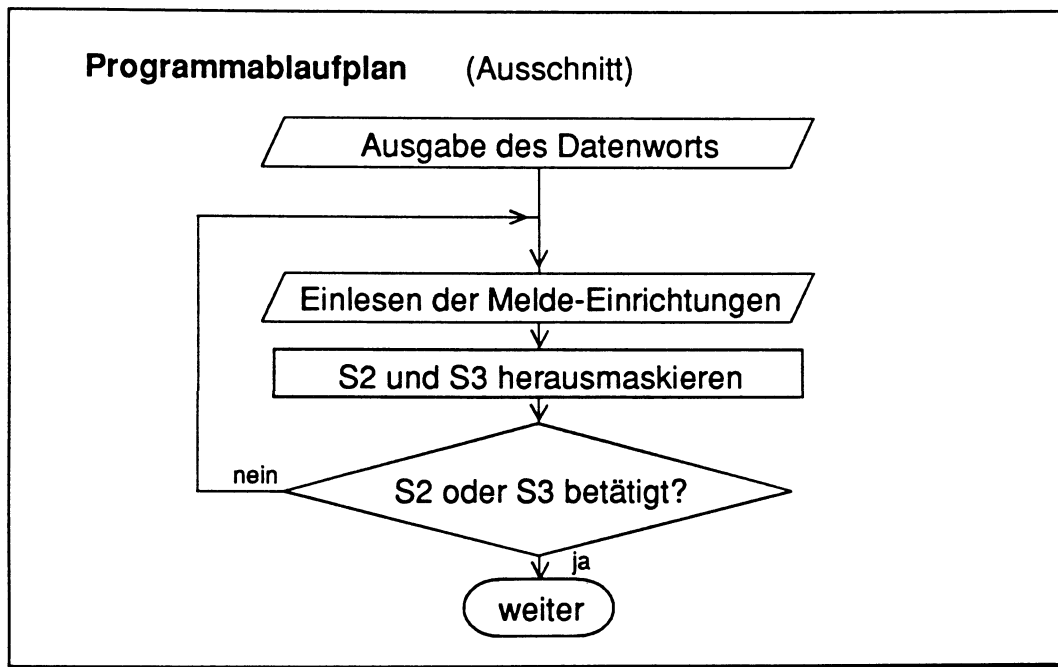
**A8d:** Das Programm ist gestartet, die Verkehrsampel 1 zeigt grün an. Warten Sie ggf. noch 24 Sekunden. Drücken Sie dann kurz die Taste S3. Was geschieht?

**A8e:** Starten Sie das Programm erneut. Gleich nach Erscheinen der Grün-Phase der Verkehrsampel 1 drücken Sie kurz S2. Was geschieht?

**A8f:** Starten Sie das Programm erneut. Gleich nach Erscheinen der Grün-Phase der Verkehrsampel 1 drücken Sie S2 und S3. Lassen Sie die Tasten erst los, wenn die Grün-Phase der jeweiligen Ampel erscheint, also das Fahrzeug fahren bzw. der Fußgänger gehen kann. Was geschieht?

**4.8.2 Erweiterte Bedarfssteuerung (BASIC)**

Die Programmierung der zeitgeführten Schritte bleibt gegenüber der Lösung der Aufgabe A6c unverändert. Nach dem zweiten Schritt, der zeitgeführt die Minstdauer der Grün-Phase der Verkehrsampel 1 garantiert, wird der zusätzliche prozeßgeführte Schritt eingeführt. Er wird durch folgenden Programmablaufplan beschrieben:



In dem vorliegenden Beispiel kann die Ausgabe des Datenworts entfallen, denn es handelt sich hier um das gleiche Datenwort wie im vorangegangenen Schritt (Grün-Phase der Verkehrsampel 1).

**Hinweis:**

Das Programm kann auch von der Diskette geladen werden (Datei A8D.ASM bzw. Datei A8D.MAT), da es vom Kursteilnehmer nicht vervollständigt werden muß. Damit wird Zeit gewonnen, um die nachfolgend beschriebenen Tests sorgfältig durchzuführen.

---

**Aufgabe A8 (BASIC)**  
**(Blatt 1 von 2)**

---

Name: 

---

---

Kurs: 

---

---

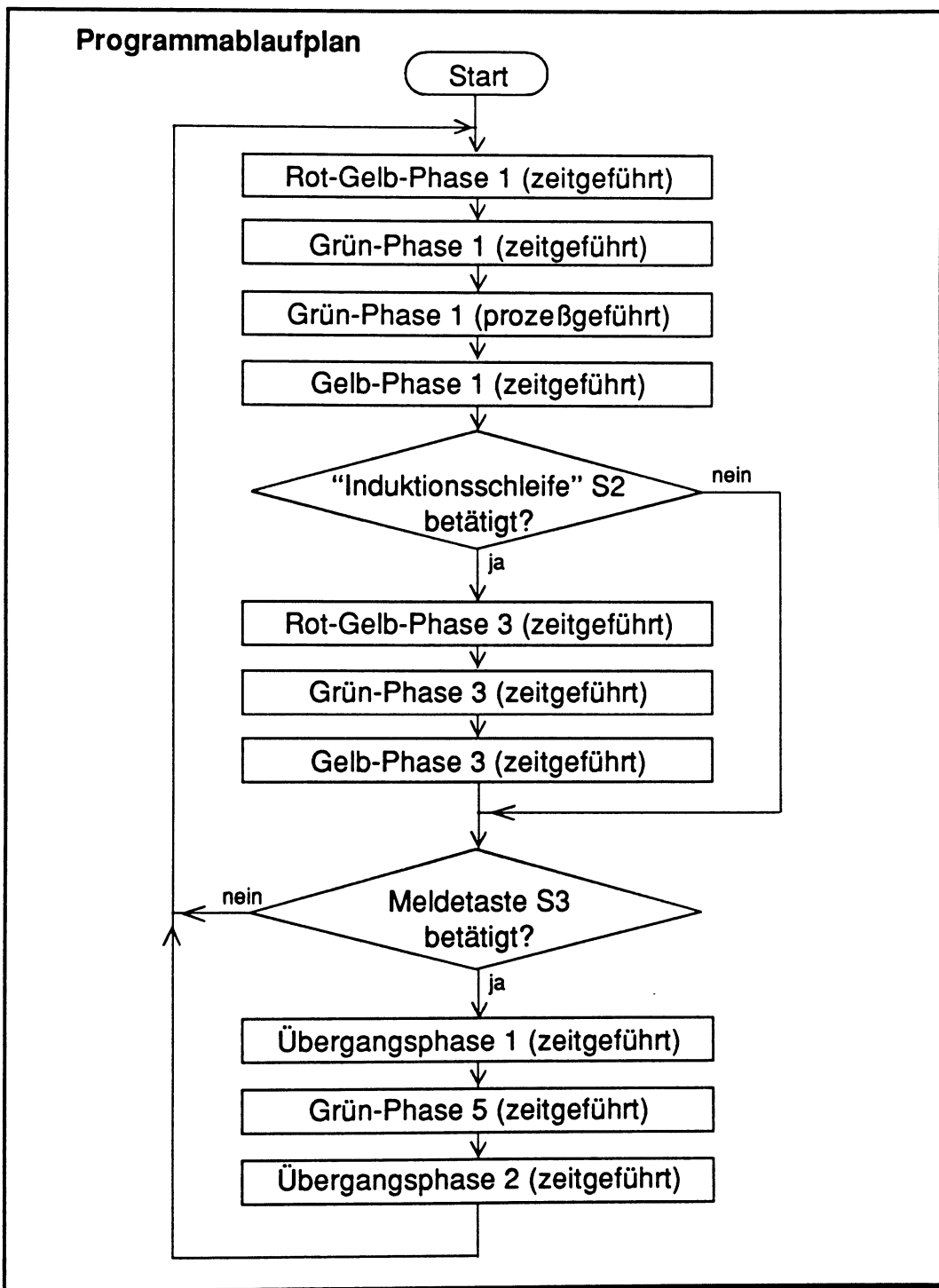
Datum: 

---

Das nachstehende BASIC-Programm steuert die Ampeln abhängig vom Verkehrsaufkommen der Nebenstraße und vom Fußgängerverkehr.

```
10 REM AUFGABE A8D
20 REM STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
30 REM BEDARFSSTEUERUNG DER SIGNALANLAGEN 3 UND 5
40 REM
50 A=0                                :REM 8-BIT-PARALLEL-AUSGABE
60 E=0                                :REM 8-BIT-PARALLEL-EINGABE
100 Z=30                              :REM 3 SEKUNDEN
110 OUT A,139                          :REM ROT1 + GELB1 + ROT3
120 GOSUB 1000
200 Z=240                             :REM 24 SEKUNDEN
210 OUT A,140                          :REM GRUEN1 + ROT3
220 GOSUB 1000
230 T=INP (E)                         :REM TASTEN EINLESEN
240 IF (T AND 6) = 0 THEN GOTO 230
300 Z=30                              :REM 3 SEKUNDEN
310 OUT A,138                          :REM GELB1 + ROT3
320 GOSUB 1000
400 IF (T AND 2) = 0 THEN GOTO 700
410 Z=30                              :REM 3 SEKUNDEN
420 OUT A,153                          :REM ROT1 + ROT3 + GELB3
430 GOSUB 1000
500 Z=240                             :REM 24 SEKUNDEN
510 OUT A,161                          :REM ROT1 + GRUEN3
520 GOSUB 1000
600 Z=30                              :REM 3 SEKUNDEN
610 OUT A,145                          :REM ROT1 + GELB3
620 GOSUB 1000
700 IF (T AND 4) = 0 THEN GOTO 100
710 Z=30                              :REM 3 SEKUNDEN
720 OUT A,137                          :REM ROT1 + ROT3
730 GOSUB 1000
800 Z=160                             :REM 16 SEKUNDEN
810 OUT A,201                          :REM ROT1 + ROT3 + GRUEN5
820 GOSUB 1000
900 Z=30                              :REM 3 SEKUNDEN
910 OUT A,137                          :REM ROT1 + ROT3
920 GOSUB 1000
999 GOTO 100
1000 REM VERZOEGERUNG (Z = ZEIT IN 1/10 SEKUNDE)
1010 FOR I=1 TO Z
1020 FOR J=1 TO 98
1030 NEXT J
1040 NEXT I
1050 RETURN
```

Die Programmverzweigung wird nachfolgend in einem Programmablaufplan dargestellt. Dieser dient der strukturellen Übersicht über das Gesamtprogramm und löst die Programmabschnitte nicht bis in die einzelnen Schritte auf.



---

**Aufgabe A8 (BASIC)**  
**(Blatt 2 von 2)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

Geben Sie das Programm ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.

Starten Sie das Programm mit dem Kommando RUN.

Beantworten Sie die folgenden Fragen. Versuchen Sie jeweils zunächst, die Antwort aus dem Verständnis des Programms zu geben. Überprüfen Sie anschließend Ihre Antwort im Experiment.

**A8d:** Das Programm ist gestartet, die Verkehrsampel 1 zeigt grün an. Warten Sie ggf. noch 24 Sekunden. Drücken Sie dann kurz die Taste S3. Was geschieht?

**A8e:** Starten Sie das Programm erneut. Gleich nach Erscheinen der Grün-Phase der Verkehrsampel 1 drücken Sie kurz S2. Was geschieht?

**A8f:** Starten Sie das Programm erneut. Gleich nach Erscheinen der Grün-Phase der Verkehrsampel 1 drücken Sie S2 und S3. Lassen Sie die Tasten erst los, wenn die Grün-Phase der jeweiligen Ampel erscheint, also das Fahrzeug fahren bzw. der Fußgänger gehen kann. Was geschieht?

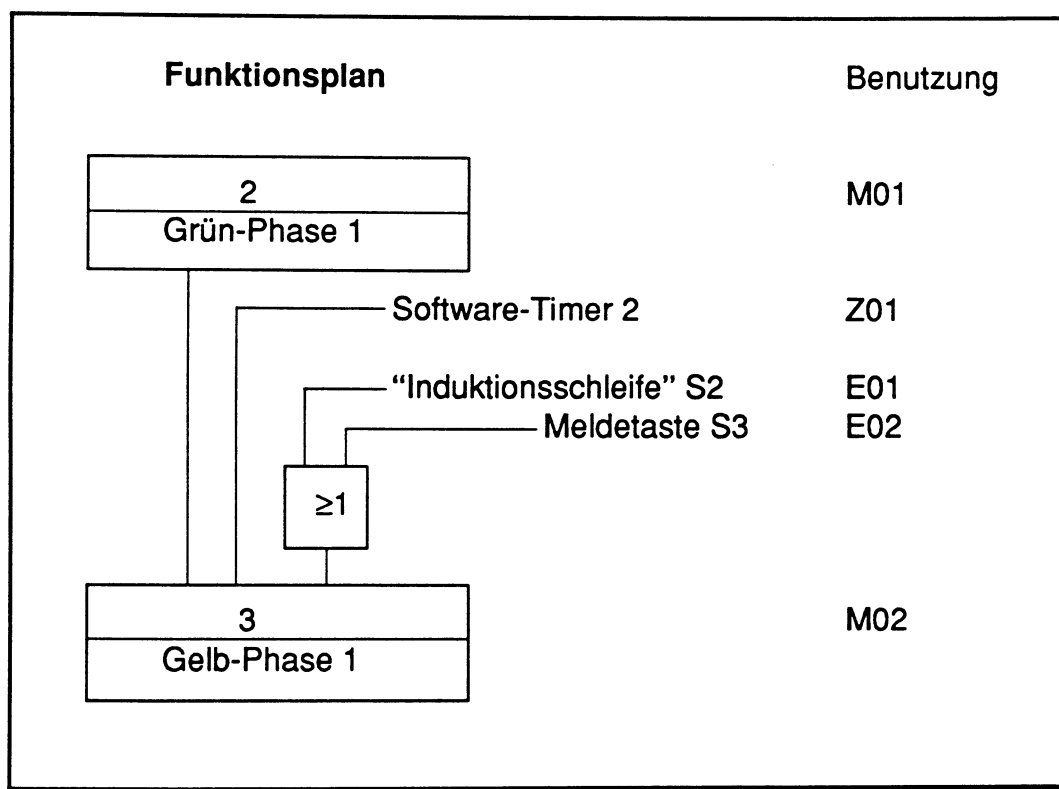


## 4.8.3 Erweiterte Bedarfssteuerung (SPS)

Die Programmierung der zeitgeführten Schritte bleibt gegenüber der Lösung der Aufgabe A6c weitgehend unverändert. Lediglich der Übergang vom zweiten zum dritten Schritt erhält eine Kombination von Bedingungen, die erfüllt sein muß:

- der Software-Timer des Schritts 2 muß abgelaufen sein,
- der Eingang E01 oder der Eingang E02 muß gesetzt sein.

Dieser Sachverhalt ist im Auszug aus dem Funktionsplan dargestellt:



Der vollständige Funktionsplan ist auf S. 114 dargestellt.

---

**Aufgabe A8 (SPS)**  
**(Blatt 1 von 4)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

Das nachstehende SPS-Programm steuert die Ampeln abhängig vom Verkehrsaufkommen der Nebenstraße und vom Fußgängerverkehr.

```
; AUFGABE A8D
; STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
; BEDARFSSTEUERUNG DER SIGNALANLAGEN 3 UND 5
; INITIALISIEREN DER ZEITGEBER
```

```
UN M 00      ; ANFANGSBEDINGUNG:
UN M 01      ; ALLE MERKER NICHT
UN M 02      ; GESETZT.
UN M 03      ;
UN M 04      ;
UN M 05      ;
UN M 06      ;
UN M 07      ;
UN M 10      ;
=L Z 00,30    ;1. ROT-GELB-PHASE AMPEL 1
=L Z 01,240   ;2. GRUEN-PHASE      AMPEL 1
=L Z 02,30    ;3. GELB-PHASE      AMPEL 1
=L Z 03,30    ;4. ROT-GELB-PHASE AMPEL 3
=L Z 04,240   ;5. GRUEN-PHASE      AMPEL 3
=L Z 05,30    ;6. GELB-PHASE      AMPEL 3
=L Z 06,30    ;7. PAUSENPHASE     AMPEL 5
=L Z 07,160   ;8. GRUENPHASE      AMPEL 5
=L Z 10,30    ;9. PAUSENPHASE     AMPEL 5
=S A 07      ; FUSSGAENGERAMPELN FREIGEBEN
=S M 00      ;
=S Z 00      ;1 STARTE PHASE 1
```

```
; ABLAUFKETTE (GESTEUERT DURCH ZEITGEBER)
```

```
U Z 00      ; PHASE 1 ABGELAUFEN
=R Z 00      ;
=R M 00      ;
=S Z 01      ; STARTE PHASE 2
=S M 01      ;2

U E 01      ; WENN FAHRZEUG AN IND.SCHL. 3
O E 02      ; ODER FUSSGAENGER AN TASTE 5
= M 23      ;3 HILFSMERKER

U M 23      ; S.O.
U Z 01      ; UND PHASE 2 ABGELAUFEN
=R Z 01      ;
=R M 01      ;
=S Z 02      ; STARTE PHASE 3
=S M 02      ;4
```

**Hinweis:** Die Kombination von UND- und ODER-Verknüpfung, die zur Gelb-Phase 1 führt, läßt sich nicht wie im Sprachgebrauch hintereinander aufschreiben. Eine solche Anweisungsliste führt zum falschen Ergebnis, da die UND-Verknüpfung Vorrang hat. Mit Hilfe der Kontaktplanausgabe wird der Fehler offensichtlich.

Die Bedingung, daß die Grün-Phase 1 beendet ist, muß in zwei UND-Verknüpfungen verwendet werden, die wiederum in einer ODER-Verknüpfung zusammengefaßt werden.

Alternativ kann auch ein Hilfsmerker verwendet werden, der das Ergebnis der ODER-Verknüpfung der beiden Eingänge zwischenspeichert und anschließend in der UND-Verknüpfung mit dem Ablauf der Grün-Phase verwendet wird.

Anweisungsliste	Kontaktplan
<b>falsch:</b>	
U E 01	I
O E 02	I-----] [-----+ I
U Z 01	I E01 I I
=R Z 01	I I I
= ....	I-----] [---] [-----+ (R)---I
	I E02 Z01 I Z01 I
	I I I
<b>richtig:</b>	
U E 01	I
U Z 01	I-----] [---] [-----+ I
O E 02	I E01 Z01 I I
U Z 01	I I I
=R Z 01	I-----] [---] [-----+ (R)---I
= ....	I E02 Z01 I Z01 I
	I I I
<b>richtig:</b>	
U E 01	I
O E 02	I-----] [-----+ I
= M 23 ;Hilfsmerker	I E01 I I
U M 23	I I I
U Z 01	I-----] [---] [-----+ ( )---I
=R Z 01	I E02 M23 I
= ....	I I I
	I-----] [---] [-----+ (R)---I
	I M23 Z01 I Z01 I
	I I I

Bei anderen SPS kann das Gleiche eventuell durch Blockbildung (Klammerung) erzielt werden:

U Z 01	I	I
U (E 01	I	I
O E 02)	I +---] [---+ I	I
=R Z 01	I I E01 I	I
= ....	I I I	I
	I-----] [---+ ] [---+-----+ (R)---I	I
	I Z01 E02 I Z01 I	I
	I I I	I

---

**Aufgabe A8 (SPS)**  
**(Blatt 2 von 4)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

```
U   E 01      ;   WENN FAHRZEUG AN IND.SCHL. 3
U   Z 02      ;   UND PHASE 3 ABGELAUFEN
=R  Z 02      ;
=R  M 02      ;
=S  Z 03      ;   STARTE PHASE 4
=S  M 03      ;5

U   E 02      ;   WENN FUSSGAENGER AN TASTE 5
UN  E 01      ;   UND KEIN FAHRZEUG AN IND.SCHL. 3
U   Z 02      ;   UND PHASE 3 ABGELAUFEN
=R  Z 02      ;
=R  M 02      ;
=S  Z 06      ;   STARTE PHASE 7
=S  M 06      ;6

U   Z 03      ;   PHASE 4 ABGELAUFEN
=R  Z 03      ;
=R  M 03      ;
=S  Z 04      ;   STARTE PHASE 5
=S  M 04      ;7

U   Z 04      ;   PHASE 5 ABGELAUFEN
=R  Z 04      ;
=R  M 04      ;
=S  Z 05      ;   STARTE PHASE 6
=S  M 05      ;8

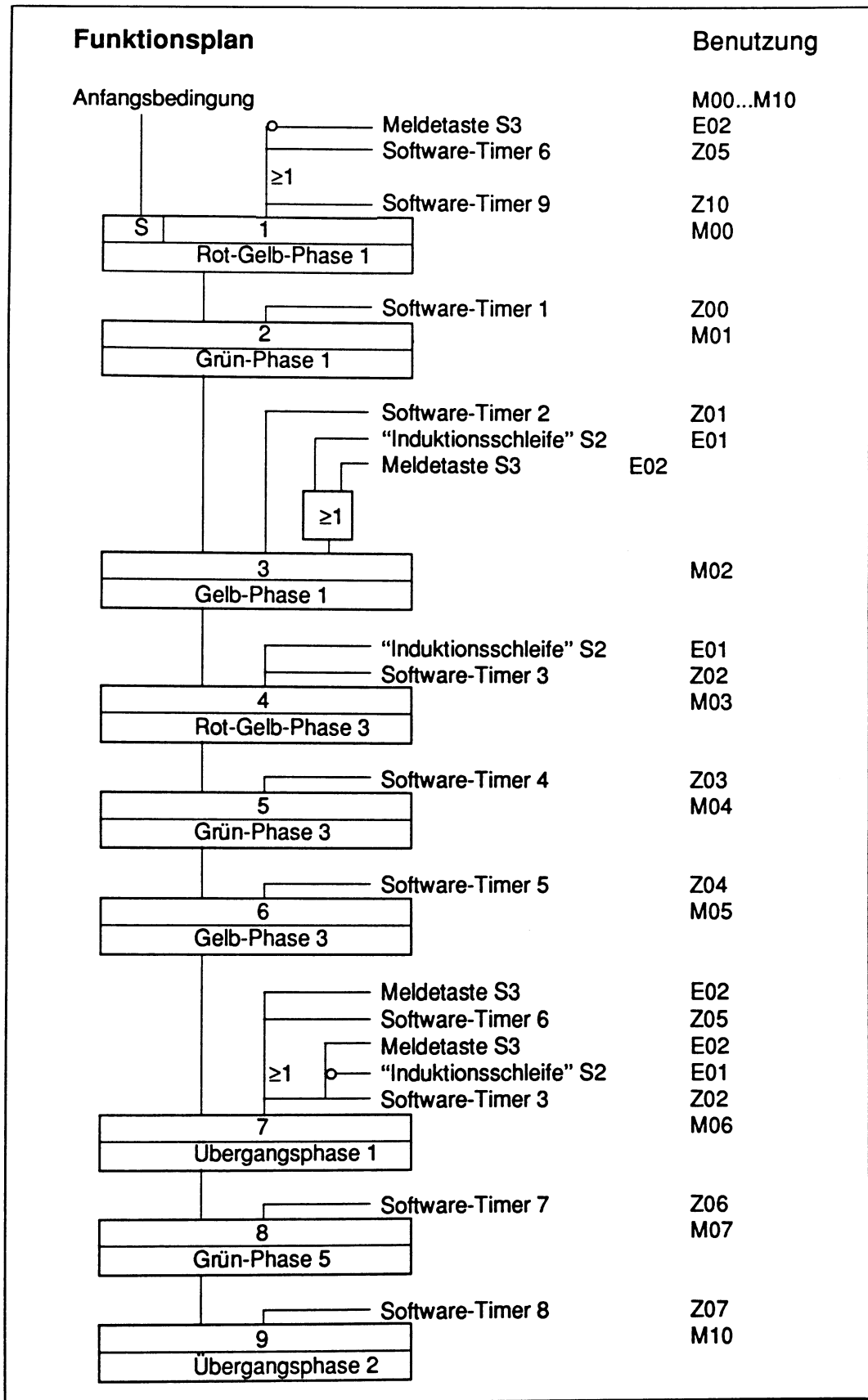
U   E 02      ;   WENN FUSSGAENGER AN TASTE 5
U   Z 05      ;   UND PHASE 6 ABGELAUFEN
=R  Z 05      ;
=R  M 05      ;
=S  Z 06      ;   STARTE PHASE 7
=S  M 06      ;9

UN  E 02      ;   WENN KEIN FUSSGAENGER AN TASTE 5
U   Z 05      ;   UND PHASE 6 ABGELAUFEN
=R  Z 05      ;
=R  M 05      ;
=S  Z 00      ;   STARTE PHASE 1
=S  M 00      ;10

U   Z 06      ;   PHASE 7 ABGELAUFEN
=R  Z 06      ;
=R  M 06      ;
=S  Z 07      ;   STARTE PHASE 8
=S  M 07      ;11

U   Z 07      ;   PHASE 8 ABGELAUFEN
=R  Z 07      ;
=R  M 07      ;
=S  Z 10      ;   STARTE PHASE 9
=S  M 10      ;12
```

---



---

**Aufgabe A8 (SPS)**  
**(Blatt 3 von 4)**

---

**Name:**

---

**Kurs:**

---

**Datum:**

---

```
U   Z 10      ;   PHASE 9 ABGELAUFEN
=R  Z 10      ;
=R  M 10      ;
=S  Z 00      ;   STARTE PHASE 1
=S  M 00      ;13
```

```
;  AUSGABESTEuerung
```

```
U   M 00      ;   IN  PHASE 1
O   M 03      ;   UND PHASE 4
O   M 04      ;   UND PHASE 5
O   M 05      ;   UND PHASE 6
O   M 06      ;   UND PHASE 7
O   M 07      ;   UND PHASE 8
O   M 10      ;   UND PHASE 9
=   A 00      ;14 ROT1 SCHALTEN
```

```
U   M 00      ;   IN  PHASE 1
O   M 02      ;   UND PHASE 3
=   A 01      ;15 GELB1 SCHALTEN
```

```
U   M 01      ;   IN  PHASE 2
=   A 02      ;16 GRUEN1 SCHALTEN
```

```
U   M 00      ;   IN  PHASE 1
O   M 01      ;   UND PHASE 2
O   M 02      ;   UND PHASE 3
O   M 03      ;   UND PHASE 4
O   M 06      ;   UND PHASE 7
O   M 07      ;   UND PHASE 8
O   M 10      ;   UND PHASE 9
=   A 03      ;17 ROT3 SCHALTEN
```

```
U   M 03      ;   IN  PHASE 4
O   M 05      ;   UND PHASE 6
=   A 04      ;18 GELB3 SCHALTEN
```

```
U   M 04      ;   IN  PHASE 5
=   A 05      ;19 GRUEN3 SCHALTEN
```

```
U   M 07      ;   IN  PHASE 8
=   A 06      ;20 GRUEN5 SCHALTEN
```

```
END
```

Geben Sie das Programm ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.

Starten Sie das Programm mit dem Kommando GO.

Die Programmverzweigung (hier z.B. nach der Gelb-Phase 1) wird durch die Formulierung zweier Übergangsbedingungen formuliert, die sich wechselseitig ausschließen. Der wechselseitige Ausschluß ist wichtig, da sonst zwei Teilschrittfolgen gleichzeitig ablaufen würden, die ggf. zu einander widersprechenden oder unsinnigen Resultaten führen würden. Diese Form der Verzweigung wird ODER-Verzweigung genannt.

In anderen Steuerungsaufgaben kann eine Verzweigung in zwei oder mehrere parallel ablaufende Teilschrittfolgen durchaus sinnvoll und erwünscht sein (sog. UND-Verzweigung).

Die beiden nachstehend aufgeführten Übergangsbedingungen stellen die Verzweigung nach der Gelb-Phase 1 dar.

**Anweisungsliste**

```
U   E 01      ;   WENN FAHRZEUG AN IND.SCHL. 3
U   Z 02      ;   UND PHASE 3 ABGELAUFEN
=R  Z 02      ;
=R  M 02      ;
=S  Z 03      ;   STARTE PHASE 4
=S  M 03      ;5

U   E 02      ;   WENN FUSSGAENGER AN TASTE 5
UN  E 01      ;   UND KEIN FAHRZEUG AN IND.SCHL. 3
U   Z 02      ;   UND PHASE 3 ABGELAUFEN
=R  Z 02      ;
=R  M 02      ;
=S  Z 06      ;   STARTE PHASE 7
=S  M 06      ;6
```

---

**Aufgabe A8 (SPS)**  
**(Blatt 4 von 4)**

---

Name: 

---

---

Kurs: 

---

---

Datum: 

---

Beantworten Sie die folgenden Fragen. Versuchen Sie jeweils zunächst, die Antwort aus dem Verständnis des Programms zu geben. Überprüfen Sie anschließend Ihre Antwort im Experiment.

**A8d:** Das Programm ist gestartet, die Verkehrsampel 1 zeigt grün an. Warten Sie ggf. noch 24 Sekunden. Drücken Sie dann die Taste S3 für mindestens fünf Sekunden. Was geschieht?

**A8e:** Starten Sie das Programm erneut. Gleich nach Erscheinen der Grün-Phase der Verkehrsampel 1 drücken Sie kurz S2. Was geschieht?

**A8f:** Starten Sie das Programm erneut. Gleich nach Erscheinen der Grün-Phase der Verkehrsampel 1 drücken Sie S2 und S3. Lassen Sie die Tasten erst los, wenn die Grün-Phase der jeweiligen Ampel erscheint, also das Fahrzeug fahren bzw. der Fußgänger gehen kann. Was geschieht?

**A8g:** Starten Sie das Programm erneut. Gleich nach Erscheinen der Grün-Phase der Verkehrsampel 1 drücken Sie die Taste S2. Lassen Sie die Taste innerhalb der Gelbphase der Verkehrsampel 1 los. Was geschieht?



**Hinweis:**

Wenn der Kursteilnehmer den BFZ-SPS-Interpreter in der Version der FTÜ benutzt, so wird von nachstehendem gleichwertigen Programm ausgegangen.

```
* /M00*/M01*/M02*/M03*/M04*/M05*/M06*/M07*/M10=LZ00,30=LZ01,240
=LZ02,30=LZ03,30=LZ04,240=LZ05,30=LZ06,30=LZ07,160=LZ10,30
=SA07=SM00=SZ00
*Z00=RZ00=RM00=SZ01=SM01
*E01+E02=M23
*M23*Z01=RZ01=RM01=SZ02=SM02
*E01*Z02=RZ02=RM02=SZ03=SM03
*E02*/E01*Z02=RZ02=RM02=SZ06=SM06
*Z03=RZ03=RM03=SZ04=SM04
*Z04=RZ04=RM04=SZ05=SM05
*E02*Z05=RZ05=RM05=SZ06=SM06
*/E02*Z05=RZ05=RM05=SZ00=SM00
*Z06=RZ06=RM06=SZ07=SM07
*Z07=RZ07=RM07=SZ10=SM10
*Z10=RZ10=RM10=SZ00=SM00
*M00+M03+M04+M05+M06+M07+M10=A00
*M00+M02=A01
*M01=A02
*M00+M01+M02+M03+M06+M07+M10=A03
*M03+M05=A04
*M04=A05
*M07=A06
```

**Hinweis:**

Die Programme können auch von der Diskette geladen werden (Dateien A8D.DOK bzw. A8D.DIN bzw. A8D.SPS), da sie vom Kursteilnehmer nicht vervollständigt werden müssen. Damit wird Zeit gewonnen, um die beschriebenen Tests sorgfältig durchführen zu können.

\_\_\_\_\_

\_\_\_\_\_

**4.9 Aufgabe 9: Speicherung der Melde-Impulse**

Die Veruche mit dem Programm aus Aufgabe A8 haben dessen Unzulänglichkeit gezeigt. Um ein korrektes Funktionieren des Programms zu gewährleisten, muß die "Induktionsschleife" S2 bzw. die Meldetaste S3 gedrückt werden, bis der entsprechende Verkehrsweg "grünes Licht" bekommt.

Nun mag zwar bei der Induktionsschleife offen sein, ob sie nur einen kurzen Impuls liefert, sobald ein Fahrzeug über ihr angehalten hat, oder einen andauernden High-Pegel. Im zweiten Fall erweist sich das Programm durchaus als brauchbar, wenn auch nicht als absolut zuverlässig (SPS-Version). Im ersten Fall ist das Programm unbrauchbar.

Bei der Melde-Einrichtung für die Fußgänger ist die Situation klar: keinem Passanten wird zuzumuten sein, solange die Meldetaste zu drücken, bis er die Fahrbahn überqueren kann. Das Programm aus Aufgabe A8 ist somit unbrauchbar.

In der folgenden Aufgabe wird dieses Manko durch Speicherung der Impulse von den Melde-Einrichtungen beseitigt.

**Hinweis:**

Fortsetzung der Aufgabe A9

- für Assembler-Programmierung auf S. 122,
- für BASIC-Programmierung auf S. 128,
- für SPS-Programmierung auf S. 132.

---

**Aufgabe A9 (Einführung)**

---

Name: \_\_\_\_\_

---

Kurs: \_\_\_\_\_

---

Datum: \_\_\_\_\_

---

**A9a:** Stellen Sie hier Ihre Kritik an dem Programm aus Aufgabe A8 zusammen. Gegebenenfalls laden Sie nochmals das Programm der Aufgabe A8 und führen Sie die Tests gemäß Aufgabenstellung A8d bis A8f (bzw. A8g) sowie eigene Tests durch.

## 4.9.1 Speicherung der Melde-Impulse (Assembler)

Um die Ampelanlage betriebssicher zu machen, müssen die Melde-Einrichtungen (nahezu) permanent überprüft werden. Es bietet sich an, eine derartige Überwachung in die Verzögerungsroutine einzubauen, da diese bis auf die kurzen Laufzeiten des Hauptprogramms praktisch dauernd aktiv ist.

Dabei wird das eingelesene Byte einer ODER-Verknüpfung mit der Speicherzelle TASTEN unterzogen. Auf diese Weise bleibt ein einmal gesetztes Bit gespeichert, auch wenn später die Taste losgelassen oder eine andere Taste gedrückt wird.

## Zeitliche Entwicklung des Signal- und Speicherzustands

Eingabe	Variable TASTEN	Ergebnis ODER	
00000000	00000000	00000000	
00000000	00000000	00000000	
00000000	00000000	00000000	
00000000	00000000	00000000	
00000010	00000000	00000010	S2 gedrückt
00000010	00000010	00000010	Impuls gespeichert
00000010	00000010	00000010	
00000010	00000010	00000010	
00000000	00000010	00000010	S2 losgelassen
00000000	00000010	00000010	
00000000	00000010	00000010	
00000000	00000010	00000010	
00000000	00000010	00000010	
00000000	00000010	00000010	
00000000	00000010	00000010	
00000100	00000010	00000110	S3 gedrückt
00000100	00000110	00000110	Impuls gespeichert
00000100	00000110	00000110	
00000100	00000110	00000110	
00000100	00000110	00000110	
00000100	00000110	00000110	
00000000	00000110	00000110	S3 losgelassen
00000000	00000110	00000110	
00000000	00000110	00000110	
↓	00000000	00000110	
Zeit	00000000	00000110	

---

**Aufgabe A8 (Assembler)**  
**(Blatt 1 von 3)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

**A9b:** Das nachstehende Assemblerprogramm steuert die Ampeln abhängig vom Verkehrsaufkommen der Nebenstraße und vom Fußgängerverkehr. Impulse von der "Induktionsschleife" S2 und der Meldetaste S3 werden gespeichert.

Das Programm ist eine Weiterentwicklung desjenigen aus Aufgabe A8. Füllen Sie die Leerstellen in dem Programm aus. Beachten Sie, daß das Registerpaar HL auf die Speicherzelle TASTEN zeigt, diese also unter Verwendung des Pseudoregisters M verwendet werden kann.

```
0000          ; AUFGABE A9B
0000          ; STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
0000          ; BEDARFSSTEUERUNG DER SIGNALANLAGEN 3 UND 5
0000          ; SPEICHERUNG VON TASTENIMPULSEN
0000          ;
0000          EINGABE EQU    0          ;8-BIT-PARALLEL-EINGABE
0000          AUSGABE EQU    0          ;8-BIT-PARALLEL-AUSGABE
0000          ;
0000          ORG      0E000    ;STARTADRESSE
E000          ;
E000 21 9AE0          LXI      H,TASTEN ;ZEIGER AUF DATENWORT
E003 01 1E00    PHASE1: LXI      B,001E ;3 SEKUNDEN
E006 3E 8B          MVI      A,8B    ;ROT1 + GELB1 + ROT3
E008 D3 00          OUT      AUSGABE
E00A CD 80E0          CALL     DELAY
E00D 01 F000    PHASE2: LXI      B,00F0 ;24 SEKUNDEN
E010 3E 8C          MVI      A,8C    ;GRUEN1 + ROT3
E012 D3 00          OUT      AUSGABE
E014 CD 80E0          CALL     DELAY
E017 01 0100    PHAS2A: LXI      B,0001 ;1/10 SEKUNDE
E01A CD 80E0          CALL     DELAY ;LIEST TASTEN
E01D 7E          MOV      A,M    ;TASTENSTELLUNGEN
E01E E6 06          ANI      06    ;TASTE 3 UND TASTE 5 MASKIEREN
E020 CA 17E0          JZ      PHAS2A ;WEITER WARTEN
E023 01 1E00    PHASE3: LXI      B,001E ;3 SEKUNDEN
E026 3E 8A          MVI      A,8A    ;GELB1 + ROT3
E028 D3 00          OUT      AUSGABE
E02A CD 80E0          CALL     DELAY
E02D 7E          PHASE4: MOV     A,M    ;TASTENSTELLUNGEN
E02E E6 02          ANI      02    ;TASTE 3 GEDRUECKT?
E030 CA 55E0          JZ      PHASE7 ;NEIN - WEITER
E033 01 1E00          LXI      B,001E ;3 SEKUNDEN
E036 3E 99          MVI      A,99    ;ROT1 + ROT3 + GELB3
E038 D3 00          OUT      AUSGABE
E03A CD 80E0          CALL     DELAY
E03D 01 F000    PHASE5: LXI      B,00F0 ;24 SEKUNDEN
E040 3E A1          MVI      A,0A1    ;ROT1 + GRUEN3
E042 D3 00          OUT      AUSGABE
E044 CD 80E0          CALL     DELAY
```

Die gesetzten Bits können wieder zurückgesetzt werden, wenn der entsprechenden Verkehrsrichtung freie Fahrt erteilt wurde. In dem nachstehenden Programm erfolgt dies zu Ende der Gelb-Phase bzw. Übergangsphase 2. Das selektive Rücksetzen von Bits wird durch eine UND-Verknüpfung mit einer Maskenkonstanten bewirkt. Im Beispiel wird der gespeicherte Impuls von S2 zurückgesetzt:

**Selektives Löschen von Bits**

TASTEN:	00000110	
Maskenkonstante:	11111101	= FD (hex)
<hr/>		
Ergebnis UND:	00000100	

**Hinweis:**

Eine Falle für den Programmierer lauert in der Verlängerung der Grün-Phase 1. Diese ist prozeßgesteuert (s. *Kapitel 4.7*) und benutzt daher nicht die Verzögerungsroutine. Impulse während dieser Phase würden somit nicht abgefragt und gingen verloren.

Es gibt mehrere Alternativen, dieses Problem zu lösen. In dem nachstehenden Programm wird in jedem Durchlauf der Abfrageschleife eine Verzögerungs-Zeitschleife von 1/10 Sekunde eingelegt, die der Speicherung der Impulse dient. Damit erfolgt das Weiterschalten in die Gelb-Phase 1 maximal um 1/10 Sekunde verzögert, was in der Praxis jedoch nicht spürbar ist.

**Hinweis:**

Die innere Schleife der Verzögerungsroutine wird jetzt weniger oft als in den vorangegangenen Aufgaben durchlaufen. Damit wird den zusätzlichen Befehlen in der Schleife Rechnung getragen. Die Zahl der Schleifendurchläufe wird wie in *Kapitel 5.1* berechnet.

# **Aufgabe A8 (Assembler)** **(Blatt 2 von 3)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

```

E047 01 1E00      PHASE6: LXI      B,001E    ;3 SEKUNDEN
E04A 3E 91        MVI      A,91      ;ROT1 + GELB3
E04C D3 00        OUT      AUSGABE
E04E CD 80E0      CALL     DELAY
E051 ..          ...      ...      ;SPEICHER ZURUECKSETZEN
E052 .. ..       ...      ...      ;BIT 1 LOESCHEN
E054 ..          ...      ...      ;WIEDER SPEICHERN
E055 7E          PHASE7: MOV     A,M      ;TASTENSTELLUNGEN
E056 E6 04        ANI      04      ;TASTE 5 GEDRUECKT?
E058 CA 03E0      JZ       PHASE1     ;NEIN - WEITER
E05B 01 1E00      LXI      B,001E    ;3 SEKUNDEN
E05E 3E 89        MVI      A,89      ;ROT1 + ROT3
E060 D3 00        OUT      AUSGABE
E062 CD 80E0      CALL     DELAY
E065 01 A000      PHASE8: LXI      B,00A0    ;16 SEKUNDEN
E068 3E C9        MVI      A,0C9     ;ROT1 + ROT3 + GRUEN5
E06A D3 00        OUT      AUSGABE
E06C CD 80E0      CALL     DELAY
E06F 01 1E00      PHASE9: LXI      B,001E    ;3 SEKUNDEN
E072 3E 89        MVI      A,89      ;ROT1 + ROT3
E074 D3 00        OUT      AUSGABE
E076 CD 80E0      CALL     DELAY
E079 ..          ...      ...      ;SPEICHER ZURUECKSETZEN
E07A .. ..       ...      ...      ;BIT 2 LOESCHEN
E07C ..          ...      ...      ;WIEDER SPEICHERN
E07D             ;
E07D C3 03E0      JMP      PHASE1     ;ENDLOSSCHLEIFE
E080             ;
E080             ;      UNTERPROGRAMM DELAY
E080 C5          DELAY: PUSH     B      ;REGISTER RETTEN
E081 D5          PUSH     D
E082 F5          PUSH     PSW
E083 11 5C11      DELAY1: LXI      D,115C    ;ZEITBASIS 1/10 S
E086 .. ..       DELAY2: ..      .....    ;TASTENSTELLUNGEN EINLESEN
E088 ..          ...      .          ;IM DATENWORT SPEICHERND
E089 ..          ...      ...      ;ABLEGEN
E08A 1B          DCX      D          ;GEGEN NULL ZAEHLEN
E08B 7A          MOV      A,D      ;TESTE AUF NULL
E08C B3          ORA      E
E08D C2 86E0      JNZ      DELAY2     ;SCHLEIFENENDE DELAY2
E090 0B          DCX      B          ;GEGEN NULL ZAEHLEN
E091 78          MOV      A,B      ;TESTE AUF NULL
E092 B1          ORA      C
E093 C2 83E0      JNZ      DELAY1     ;SCHLEIFENENDE DELAY1
E096 F1          POP      PSW
E097 D1          POP      D
E098 C1          POP      B
E099 C9          RET              ;UNTERPROGRAMM DELAY
E09A             ;
E09A             ; DATENSPEICHER
E09A 00          TASTEN: DB      0      ;SPEICHER FUER EINGABEBYTE

```



---

## 4. Aufgaben

---

---

**Aufgabe A8 (Assembler)**  
**(Blatt 3 von 3)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

Geben Sie das Programm mit Hilfe des Memory- oder Assembler-Kommandos des MAT 85 ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des Print- oder Disassembler-Kommandos.

Starten Sie das Programm mit dem GO-Kommando.

**A9c:** Prüfen Sie das Programm, indem Sie zu verschiedenen Zeitpunkten die Melde-Einrichtungen S2 bzw. S3 betätigen.

Erstellen Sie ein Protokoll Ihrer Tests.

## 4.9.2 Speicherung der Melde-Impulse (BASIC)

Um die Ampelanlage betriebssicher zu machen, müssen die Melde-Einrichtungen (nahezu) permanent überprüft werden. Es bietet sich an, eine derartige Überwachung in die Verzögerungsroutine einzubauen, da diese bis auf die kurzen Laufzeiten des Hauptprogramms praktisch dauernd aktiv ist.

Dabei wird das eingelesene Byte einer ODER-Verknüpfung mit der Variablen T (für Tasten) unterzogen. Auf diese Weise bleibt ein einmal gesetztes Bit gespeichert, auch wenn später die Taste losgelassen oder eine andere Taste gedrückt wird.

## Zeitliche Entwicklung des Signal- und Speicherzustands

Eingabe	Variable T	Ergebnis ODER	
00000000	00000000	00000000	
00000000	00000000	00000000	
00000000	00000000	00000000	
00000000	00000000	00000000	
00000010	00000000	00000010	S2 gedrückt
00000010	00000010	00000010	Impuls gespeichert
00000010	00000010	00000010	
00000010	00000010	00000010	
00000000	00000010	00000010	S2 losgelassen
00000000	00000010	00000010	
00000000	00000010	00000010	
00000000	00000010	00000010	
00000000	00000010	00000010	
00000000	00000010	00000010	
00000000	00000010	00000010	
00000100	00000010	00000110	S3 gedrückt
00000100	00000110	00000110	Impuls gespeichert
00000100	00000110	00000110	
00000100	00000110	00000110	
00000100	00000110	00000110	
00000000	00000110	00000110	S3 losgelassen
00000000	00000110	00000110	
00000000	00000110	00000110	
↓	00000000	00000110	
Zeit	00000000	00000110	

---

**Aufgabe A8 (BASIC)**  
**(Blatt 1 von 2)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

**A9b:** Das nachstehende BASIC-Programm steuert die Ampeln abhängig vom Verkehrsaufkommen der Nebenstraße und vom Fußgängerverkehr. Impulse von der "Induktionsschleife" S2 und der Meldetaste S3 werden gespeichert.

Das Programm ist eine Weiterentwicklung desjenigen aus Aufgabe A8. Füllen Sie die Leerstellen im Programm aus.

```
10 REM AUFGABE A9B
20 REM STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
30 REM BEDARFSSTEUERUNG DER SIGNALANLAGEN 3 UND 5
40 REM SPEICHERUNG VON TASTENIMPULSEN
50 A=0                                :REM 8-BIT-PARALLEL-AUSGABE
60 E=0                                :REM 8-BIT-PARALLEL-EINGABE
100 Z=30                              :REM 3 SEKUNDEN
110 OUT A,139                         :REM ROT1 + GELB1 + ROT3
120 GOSUB 1000
200 Z=240                             :REM 24 SEKUNDEN
210 OUT A,140                         :REM GRUEN1 + ROT3
220 GOSUB 1000
230 Z=1                               :REM 1/10 SEKUNDE
240 GOSUB 1000
250 IF (T AND 6) = 0 THEN GOTO 230
300 Z=30                              :REM 3 SEKUNDEN
310 OUT A,138                         :REM GELB1 + ROT3
320 GOSUB 1000
400 IF (T AND 2) = 0 THEN GOTO 700
410 Z=30                              :REM 3 SEKUNDEN
420 OUT A,153                         :REM ROT1 + ROT3 + GELB3
430 GOSUB 1000
500 Z=240                             :REM 24 SEKUNDEN
510 OUT A,161                         :REM ROT1 + GRUEN3
520 GOSUB 1000
600 Z=30                              :REM 3 SEKUNDEN
610 OUT A,145                         :REM ROT1 + GELB3
620 GOSUB 1000
630 .=. ... ..                       :REM BIT 1 LOESCHEN
700 IF (T AND 4) = 0 THEN GOTO 100
710 Z=30                              :REM 3 SEKUNDEN
720 OUT A,137                         :REM ROT1 + ROT3
730 GOSUB 1000
800 Z=160                             :REM 16 SEKUNDEN
810 OUT A,201                         :REM ROT1 + ROT3 + GRUEN5
820 GOSUB 1000
900 Z=30                              :REM 3 SEKUNDEN
910 OUT A,137                         :REM ROT1 + ROT3
920 GOSUB 1000
930 .=. ... ..                       :REM BIT 2 LOESCHEN
999 GOTO 100
```

Die gesetzten Bits können wieder zurückgesetzt werden, wenn der entsprechenden Verkehrsrichtung freie Fahrt erteilt wurde. In dem nachstehenden Programm erfolgt dies zu Ende der Gelb-Phase bzw. Übergangsphase 2. Das selektive Rücksetzen von Bits wird durch eine UND-Verknüpfung mit einer Maskenkonstanten bewirkt. Im Beispiel wird der gespeicherte Impuls von S2 zurückgesetzt:

**Selektives Löschen von Bits**

T:	00000110	
Maskenkonstante:	11111101	= 253
<hr/>		
Ergebnis UND:	00000100	

**Hinweis:**

Eine Falle für den Programmierer lauert in der Verlängerung der Grün-Phase 1. Diese ist prozeßgesteuert (s. *Kapitel 4.7*) und benutzt daher nicht die Verzögerungsroutine. Impulse während dieser Phase würden somit nicht abgefragt und gingen verloren.

Es gibt mehrere Alternativen, dieses Problem zu lösen. In der nachstehenden Lösung wird in jedem Durchlauf der Abfrageschleife eine Verzögerungs-Zeitschleife von 1/10 Sekunde eingelegt, die der Speicherung der Impulse dient. Damit erfolgt das Weiterschalten in die Gelb-Phase 1 maximal um 1/10 Sekunde verzögert, was in der Praxis jedoch nicht spürbar ist.

**Hinweis:**

Die innere Schleife der Verzögerungsroutine wird jetzt weniger oft als in den vorangegangenen Aufgaben durchlaufen. Damit wird den zusätzlichen Befehlen in der Schleife Rechnung getragen. Die Zahl der Schleifendurchläufe wird wie in *Kapitel 5.1* experimentell ermittelt.

---

**Aufgabe A8 (BASIC)**  
**(Blatt 2 von 2)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

```
1000 REM VERZOEGERUNG (Z = ZEIT IN 1/10 SEKUNDE)
1010 REM UND SPEICHERN VON TASTENBETAETIGUNGEN
1020 FOR I=1 TO Z
1030 FOR J=1 TO 7
1040 .=. . . . . :REM TASTEN LESEN UND SPEICHERN
1050 NEXT J
1060 NEXT I
1070 RETURN
```

Geben Sie das Programm ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.

Starten Sie das Programm mit dem Kommando RUN.

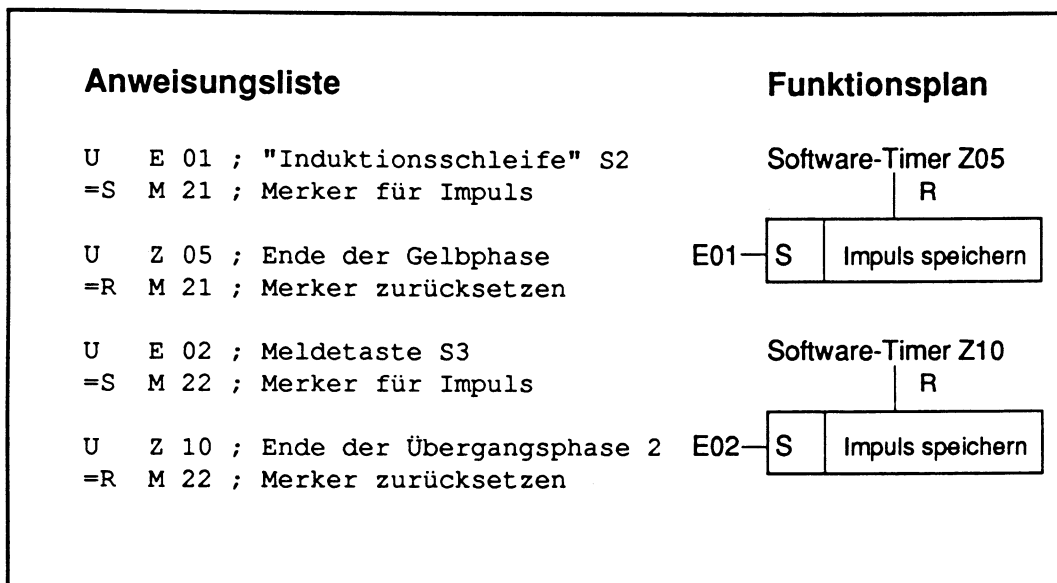
**A9c:** Prüfen Sie das Programm, indem Sie zu verschiedenen Zeitpunkten die Melde-Einrichtungen S2 bzw. S3 betätigen.

Erstellen Sie ein Protokoll Ihrer Tests.

### 4.9.3 Speicherung der Melde-Impulse (SPS)

Um die Ampelanlage betriebssicher zu machen, müssen die Impulse der Melde-Einrichtungen gespeichert werden. Sie werden dazu speichernd in Merker kopiert. Auf den Inhalt dieser Merker kann zu späteren Zeitpunkten des Steuerungsablaufs zugegriffen werden, auch wenn die Impulse von den Melde-Einrichtungen bereits wieder beendet sind.

Die Merker können wieder zurückgesetzt werden, wenn der entsprechenden Verkehrsrichtung freie Fahrt erteilt wurde. Im nachstehenden Programm erfolgt dies zu Ende der Gelb-Phase bzw. Übergangsphase 2.



---

**Aufgabe A8 (SPS)**  
**(Blatt 1 von 4)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

**A9b:** Das nachstehende SPS-Programm steuert die Ampeln abhängig vom Verkehrsaufkommen der Nebenstraße und vom Fußgängerverkehr. Impulse von der "Induktionsschleife" S2 und der Meldetaste S3 werden in den Merkern M21 bzw. M22 gespeichert.  
Das Programm ist eine Weiterentwicklung desjenigen aus Aufgabe A8. Füllen Sie die Leerstellen im Programm aus.

```
; AUFGABE A9B
; STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
; BEDARFSSTEUERUNG DER SIGNALANLAGEN 3 UND 5
; SPEICHERUNG DER TASTENIMPULSE
; INITIALISIEREN DER ZEITGEBER
```

```
UN M 00      ; ANFANGSBEDINGUNG:
UN M 01      ; ALLE MERKER NICHT
UN M 02      ; GESETZT.
UN M 03      ;
UN M 04      ;
UN M 05      ;
UN M 06      ;
UN M 07      ;
UN M 10      ;
=L Z 00,30    ;1. ROT-GELB-PHASE AMPEL 1
=L Z 01,240   ;2. GRUEN-PHASE    AMPEL 1
=L Z 02,30    ;3. GELB-PHASE     AMPEL 1
=L Z 03,30    ;4. ROT-GELB-PHASE AMPEL 3
=L Z 04,240   ;5. GRUEN-PHASE    AMPEL 3
=L Z 05,30    ;6. GELB-PHASE     AMPEL 3
=L Z 06,30    ;7. PAUSENPHASE    AMPEL 5
=L Z 07,160   ;8. GRUENPHASE     AMPEL 5
=L Z 10,30    ;9. PAUSENPHASE    AMPEL 5
=S A 07      ; FUSSGAENGERAMPELN FREIGEBEN
=S M 00      ;
=S Z 00      ;1 STARTE PHASE 1
```

```
; ABLAUFKETTE (GESTEUERT DURCH ZEITGEBER)
```

```
U   Z 00      ; PHASE 1 ABGELAUFEN
=R  Z 00      ;
=R  M 00      ;
=S  Z 01      ; STARTE PHASE 2
=S  M 01      ;2

U   M 21      ; WENN FAHRZEUG AN IND.SCHL. 3
O   M 22      ; ODER FUSSGAENGER AN TASTE 5
=   M 23      ;3 HILFSMERKER
```



**Hinweis:**

Wenn der Kursteilnehmer den BFZ-SPS-Interpreter in der Version der FTÜ benutzt, so wird von nachstehendem gleichwertigen Programm ausgegangen.

```
*/M00*/M01*/M02*/M03*/M04*/M05*/M06*/M07*/M10=LZ00,30=LZ01,240
=LZ02,30=LZ03,30=LZ04,240=LZ05,30=LZ06,30=LZ07,160=LZ10,30
=SA07=SM00=SZ00
*Z00=RZ00=RM00=SZ01=SM01
*M21+M22=M23
*M23*Z01=RZ01=RM01=SZ02=SM02
*M21*Z02=RZ02=RM02=SZ03=SM03
*M22*/M21*Z02=RZ02=RM02=SZ06=SM06
*Z03=RZ03=RM03=SZ04=SM04
*Z04=RZ04=RM04=SZ05=SM05
.....
*M22*Z05=RZ05=RM05=SZ06=SM06
*/M22*Z05=RZ05=RM05=SZ00=SM00
*Z06=RZ06=RM06=SZ07=SM07
*Z07=RZ07=RM07=SZ10=SM10
.....
*Z10=RZ10=RM10=SZ00=SM00
.....
.....
*M00+M03+M04+M05+M06+M07+M10=A00
*M00+M02=A01
*M01=A02
*M00+M01+M02+M03+M06+M07+M10=A03
*M03+M05=A04
*M04=A05
*M07=A06
```

---

**Aufgabe A8 (SPS)**  
**(Blatt 2 von 4)**

---

**Name:**

---

**Kurs:**

---

**Datum:**

---

```
U   M 23      ;   S.O.
U   Z 01      ;   UND PHASE 2 ABGELAUFEN
=R  Z 01      ;
=R  M 01      ;
=S  Z 02      ;   STARTE PHASE 3
=S  M 02      ;4

U   M 21      ;   WENN FAHRZEUG AN IND.SCHL. 3
U   Z 02      ;   UND PHASE 3 ABGELAUFEN
=R  Z 02      ;
=R  M 02      ;
=S  Z 03      ;   STARTE PHASE 4
=S  M 03      ;5

U   M 22      ;   WENN FUSSGAENGER AN TASTE 5
UN  M 21      ;   UND KEIN FAHRZEUG AN IND.SCHL. 3
U   Z 02      ;   UND PHASE 3 ABGELAUFEN
=R  Z 02      ;
=R  M 02      ;
=S  Z 06      ;   STARTE PHASE 7
=S  M 06      ;6

U   Z 03      ;   PHASE 4 ABGELAUFEN
=R  Z 03      ;
=R  M 03      ;
=S  Z 04      ;   STARTE PHASE 5
=S  M 04      ;7

U   Z 04      ;   PHASE 5 ABGELAUFEN
=R  Z 04      ;
=R  M 04      ;
=S  Z 05      ;   STARTE PHASE 6
=S  M 05      ;8

.   .  ..     ;   WENN PHASE 6 ABGELAUFEN
..  .  ..     ;9  RUECKSETZE IMPULSSPEICHER DER IND.SCHL. 3

U   M 22      ;   WENN FUSSGAENGER AN TASTE 5
U   Z 05      ;   UND PHASE 6 ABGELAUFEN
=R  Z 05      ;
=R  M 05      ;
=S  Z 06      ;   UND STARTE PHASE 7
=S  M 06      ;10

UN  M 22      ;   WENN KEIN FUSSGAENGER AN TASTE 5
U   Z 05      ;   UND PHASE 6 ABGELAUFEN
=R  Z 05      ;
=R  M 05      ;
=S  Z 00      ;   STARTE PHASE 1
=S  M 00      ;11
```

---

## 4. Aufgaben

---

---

**Aufgabe A8 (SPS)**  
**(Blatt 3 von 4)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

```
U   Z 06      ;   PHASE 7 ABGELAUFEN
=R  Z 06      ;
=R  M 06      ;
=S  Z 07      ;   STARTE PHASE 8
=S  M 07      ;12

U   Z 07      ;   PHASE 8 ABGELAUFEN
=R  Z 07      ;
=R  M 07      ;
=S  Z 10      ;   STARTE PHASE 9
=S  M 10      ;13

.   .  ..     ;   WENN PHASE 9 ABGELAUFEN
..  .  ..     ;14 RUECKSETZE IMPULSSPEICHER TASTE 5

U   Z 10      ;   PHASE 9 ABGELAUFEN
=R  Z 10      ;
=R  M 10      ;
=S  Z 00      ;   STARTE PHASE 1
=S  M 00      ;15

;   SPEICHERUNG DER EINGABEIMPULSE

.   .  ..     ;   IMPULS DER INDUKTIONSSCHLEIFE 3
..  .  ..     ;16 IN MERKER 21 SPEICHERN

.   .  ..     ;   IMPULS DER FUSSGAENGERTASTE 5
..  .  ..     ;17 IN MERKER 22 SPEICHERN

;   AUSGABESTEuerung

U   M 00      ;   IN PHASE 1
O   M 03      ;   UND PHASE 4
O   M 04      ;   UND PHASE 5
O   M 05      ;   UND PHASE 6
O   M 06      ;   UND PHASE 7
O   M 07      ;   UND PHASE 8
O   M 10      ;   UND PHASE 9
=   A 00      ;18 ROT1 SCHALTEN

U   M 00      ;   IN PHASE 1
O   M 02      ;   UND PHASE 3
=   A 01      ;19 GELB1 SCHALTEN

U   M 01      ;   IN PHASE 2
=   A 02      ;20 GRUEN1 SCHALTEN

U   M 00      ;   IN PHASE 1
O   M 01      ;   UND PHASE 2
O   M 02      ;   UND PHASE 3
O   M 03      ;   UND PHASE 4
```

---

#### **4. Aufgaben**

---

---

**Aufgabe A8 (SPS)**  
**(Blatt 4 von 4)**

---

Name: 

---

---

Kurs: 

---

---

Datum: 

---

```
O  M 06      ;  UND PHASE 7
O  M 07      ;  UND PHASE 8
O  M 10      ;  UND PHASE 9
=  A 03      ;21 ROT3 SCHALTEN

U  M 03      ;  IN  PHASE 4
O  M 05      ;  UND PHASE 6
=  A 04      ;22 GELB3 SCHALTEN

U  M 04      ;  IN PHASE 5
=  A 05      ;23 GRUEN3 SCHALTEN

U  M 07      ;  IN PHASE 8
=  A 06      ;22 GRUEN5 SCHALTEN
```

END

Geben Sie das Programm ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.

Starten Sie das Programm mit dem Kommando GO.

**A9c:** Prüfen Sie das Programm, indem Sie zu verschiedenen Zeitpunkten die Meldeeinrichtungen S2 bzw. S3 betätigen.

Erstellen Sie ein Protokoll Ihrer Tests.

**4.10 Aufgabe 10: Verkehrsdichten-abhängige Steuerung**

Die beiden Verkehrsampeln 1 und 3 sowie die Fußgängerampel 5 sollen entsprechend dem Verkehrsaufkommen gesteuert werden. Der Einfachheit halber wird zwischen drei verschiedenen Verkehrsdichten unterschieden:

- a) Kein Verkehrsaufkommen: Wenn kein Signal von der Melde-Einrichtung vorliegt, besteht auch keine Notwendigkeit, dem betreffenden Verkehrsweg freie Bahn einzuräumen.
- b) Normales Verkehrsaufkommen: Wenn ein Signal von der betreffenden Melde-Einrichtung vorliegt (und kein hohes Verkehrsaufkommen laut Punkt c registriert wurde), wird dem betreffenden Verkehrsweg je eine Rot-Gelb-, Grün- und Gelb-Phase eingeräumt. Die Grün-Phase bleibt mindestens für die einfache Mindestdauer bestehen. Sie kann auch beliebig länger andauern, solange kein Verkehrsaufkommen von den beiden anderen Verkehrswegen gemeldet wird.
- c) Hohes Verkehrsaufkommen: Wenn gleich nach dem Sperren eines Verkehrswegs (innerhalb von 5 Sekunden nach Ende der Gelb-Phase) erneut ein Signal der dazugehörigen Melde-Einrichtung vorliegt, besteht offensichtlich ein kontinuierlicher Verkehrsfluß. In diesem Fall wird hohe Verkehrsdichte registriert. Die Grün-Phase bleibt mindestens für die doppelte Mindestdauer bestehen. Sie kann auch beliebig länger andauern, solange kein Verkehrsaufkommen von den beiden anderen Verkehrswegen gemeldet wird.

Wenn nach dem Ende der Gelb-Phase eines Verkehrswegs Meldungen beider anderer Verkehrswege vorliegen, werden die Verkehrswege im Uhrzeigersinne freigegeben (1 - 3 - 5).

Die Fußgängerampeln 2 und 4 sind weiterhin starr an die Grün-Phasen der Verkehrsampeln 1 und 3 gekoppelt.

Als Zeitkonstanten werden für die Steuerung festgesetzt:

Mindestdauer der Grün-Phase der Verkehrsampel 1: 16 bzw. 32 s,  
Mindestdauer der Grün-Phase der Verkehrsampel 2: 10 bzw. 20 s,  
Mindestdauer der Grün-Phase der Fußgängerampel 5: 10 bzw. 20 s,  
Dauer der Rot-Gelb-, Gelb- und Übergangsphasen: 3 s,  
Kriterium für hohe Verkehrsdichte: Meldesignal innerhalb von 5 Sekunden nach Beginn der Rot-Phase.

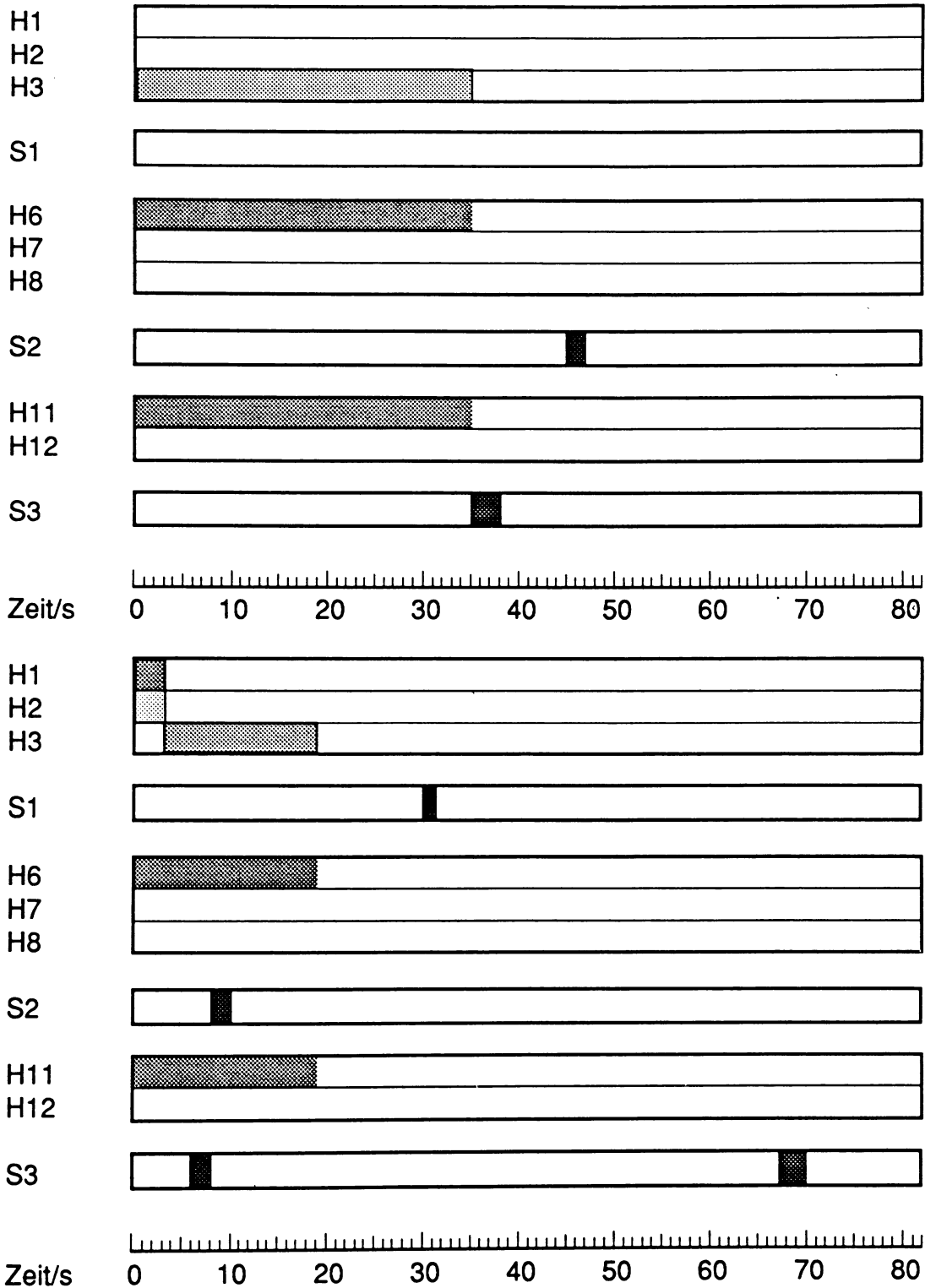
**Aufgabe A10 (Einführung)**  
(Blatt 1 von 2)

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

**A10a:** Vervollständigen Sie folgende Signal-Zeit-Diagramme entsprechend der Funktionsbeschreibung der Ampelsteuerung:





**Anmerkung:**

Im Falle hoher Verkehrsdichte wird der entsprechende Verkehrsweg aufgrund der verlängerten Grün-Phase bevorzugt behandelt. Tritt jedoch hohe Verkehrsdichte an allen drei Verkehrswegen auf, so stehen die Zeitdauern der Grün-Phasen wieder im gleichen Verhältnis. Dennoch wird der Durchsatz durch die Kreuzung effizienter, da die Zeiten der Rot-Gelb- und Gelb-Phasen sowie Anfahrvorgänge der Fahrzeuge angesichts der längeren Grün-Phasen nicht mehr so sehr ins Gewicht fallen.

**Hinweis:**

Fortsetzung der Aufgabe A10

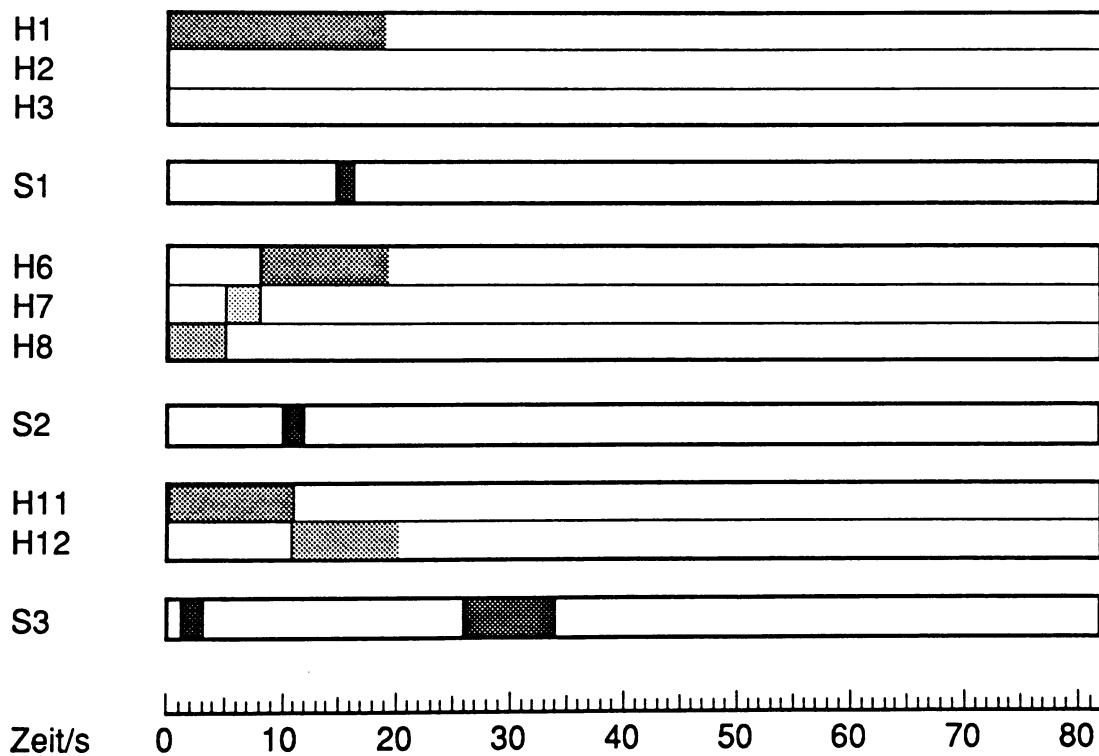
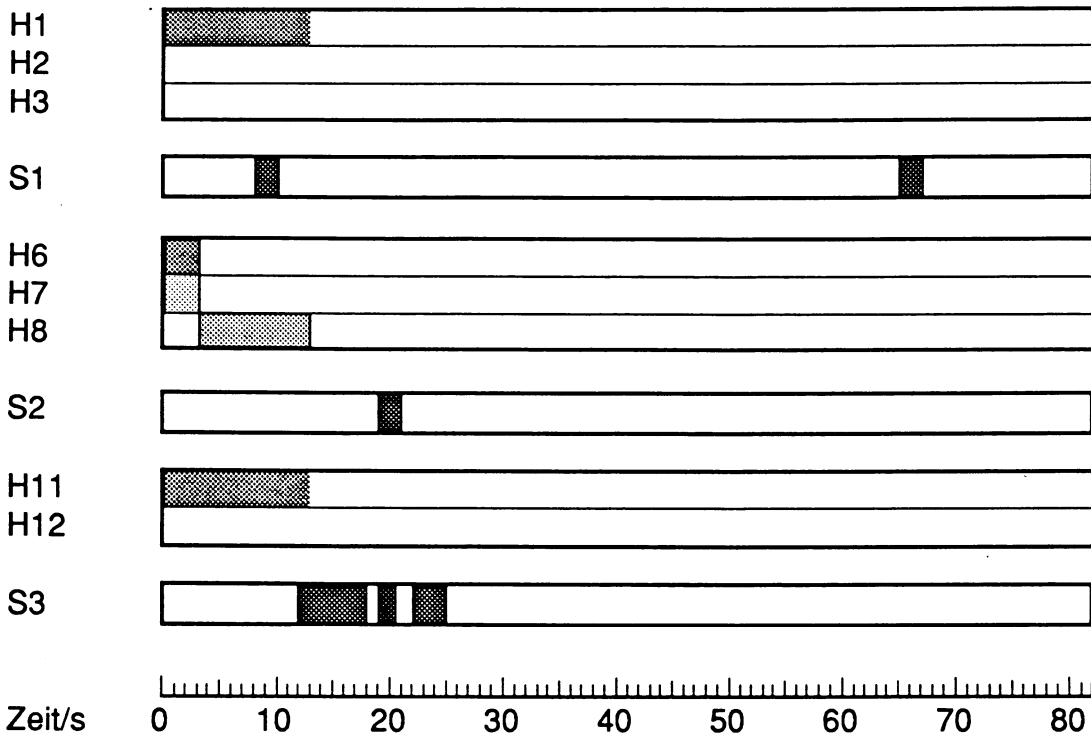
- für Assembler-Programmierung auf S.144,
- für BASIC-Programmierung auf S. 158,
- für SPS-Programmierung auf S. 164.

**Aufgabe A10 (Einführung)**  
(Blatt 2 von 2)

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

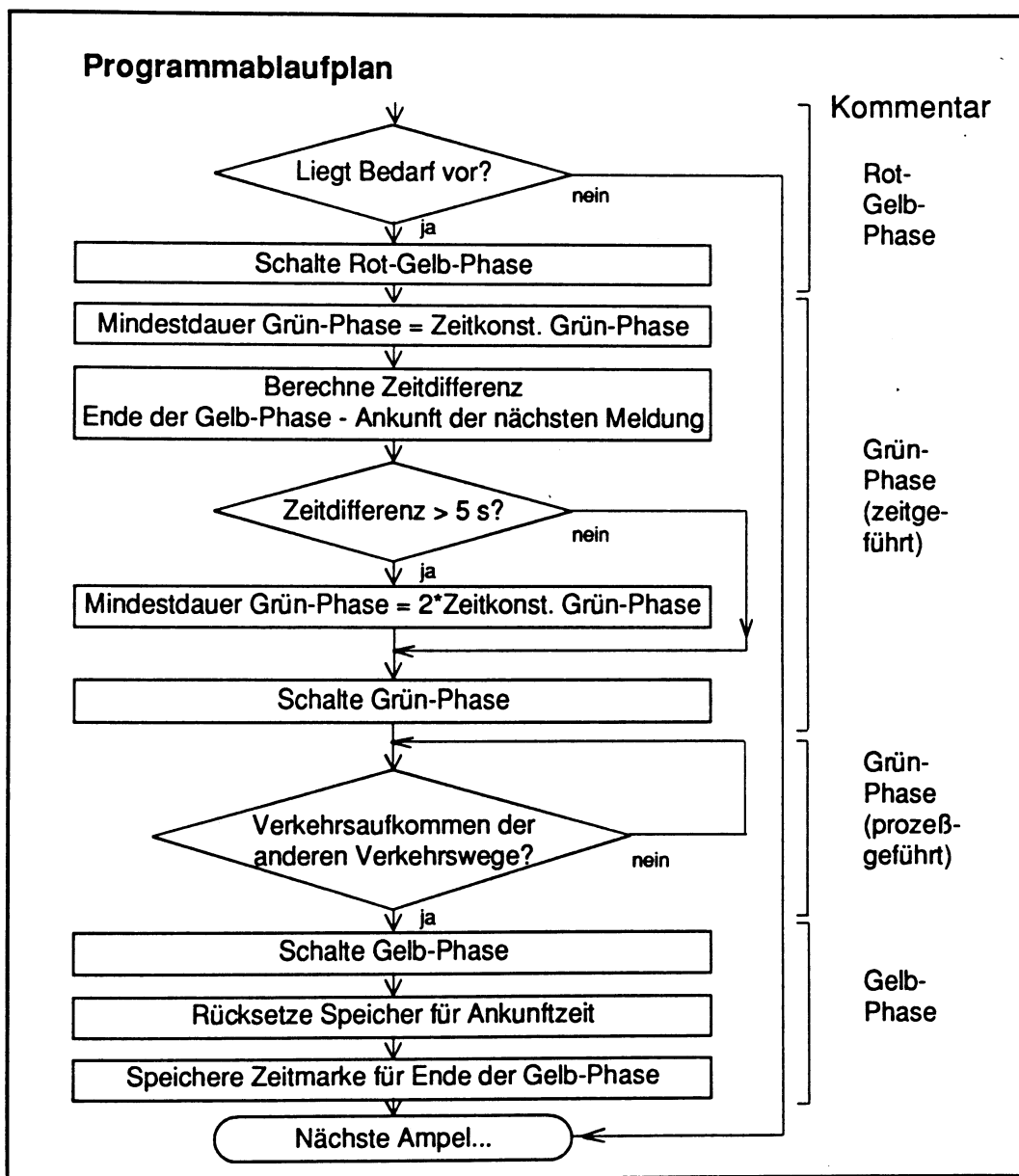
Datum: \_\_\_\_\_



4.10.1 Steuerungsprogramm mit Berücksichtigung der Verkehrsdichte  
(Assembler)

Im Gegensatz zu den Aufgaben A7, A8 und A9 ist die Behandlung der drei Verkehrswege im folgenden Steuerungsprogramm vollkommen gleich - lediglich die eingestellten Zeitkonstanten geben der Hauptstraße einen Vorzug.

Es genügt daher, den Programmablaufplan für die Steuerung einer Ampel darzustellen. Die beiden anderen Ampeln schließen sich in gleicher Form an, und am Ende steht ein Sprung an den Anfang, um den ganzen Vorgang zyklisch zu wiederholen.



**Aufgabe A10 (Assembler)**  
(Blatt 1 von 7)

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

Das nachstehende Assemblerprogramm führt die Steuerung der Ampeln in Abhängigkeit von der Verkehrsdichte durch.

```
0000      ; AUFGABE A10B, TEIL 1
0000      ; STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
0000      ; ENTSPRECHEND DER VERKEHRSFLUSSDICHT
0000      ;
0000      ;*****
0000      ;* ACHTUNG! STARTADRESSE E008 ! *
0000      ;*****
0000      ;
0000      ; STRATEGIE:
0000      ; WENN INNERHALB EINES KOMPLETTEN ZYKLUS DIE
0000      ; BEDARFSTASTE BZW. INDUKTIONSSCHLEIFE NICHT
0000      ; AKTIVIERT WURDE, WERDEN DIE PHASEN FUER DIESEN
0000      ; VERKEHRSWEG AUSGELASSEN.
0000      ; WENN INNERHALB VON 5 SEKUNDEN NACH DER GRUEN-
0000      ; PHASE DIE BEDARFSTASTE BZW. DIE INDUKTIONSSCHLEIFE
0000      ; AKTIVIERT WURDE, LIEGT EINE HOHE VERKEHRSDICHTE
0000      ; VOR. DIE GRUENPHASE DIESES VERKEHRSWEGS WIRD VER-
0000      ; DOPPELT.
0000      ;
0000      ; KONSTANTEN:
0000      G1      EQU      0096      ;GRUEN-PHASE 1 = 15 S
0000      G3      EQU      0064      ;GRUEN-PHASE 3 = 10 S
0000      G5      EQU      0064      ;GRUEN-PHASE 5 = 10 S
0000      ZW      EQU      001E      ;UEBERGANGSPHASEN = 3 S
0000      ZEITK   EQU      0A25      ;ZEITKONSTANTE 1/10 S
0000      ;
0000      ; DATENBEREICH:
0000      TASTEN  EQU      0E18B      ;TASTENBYTE
0000      AN1     EQU      0E18C      ;ANKUNFTZEIT 1
0000      MARKE1  EQU      0E18E      ;ENDE LETZTE GRUENPHASE 1
0000      AN3     EQU      0E190      ;ANKUNFTZEIT 3
0000      MARKE3  EQU      0E192      ;ENDE LETZTE GRUENPHASE 3
0000      AN5     EQU      0E194      ;ANKUNFTZEIT 5
0000      MARKE5  EQU      0E196      ;ENDE LETZTE GRUENPHASE 5
0000      ZEIT    EQU      0E198      ;UHRZEIT
0000      ;
0000      ; ADRESSEN AUS TEIL 2
0000      PHASE7  EQU      0E0B6
0000      DELAY   EQU      0E114      ;VERZOEGERUNGSRoutine
0000      SUB2     EQU      0E17C      ;16-BIT-SUBTRAKTIONSRoutine
0000      BMAL2   EQU      0E183      ;16-BIT-LINKSSCHIEBEN
0000      ;
0000      ; EIN- UND AUSGABE:
0000      EINGABE EQU      0          ;8-BIT-PARALLEL-EINGABE
0000      AUSGABE EQU      0          ;8-BIT-PARALLEL-AUSGABE
0000      ;
0000      ORG      0E000      ;STARTADRESSE
E000      ;
```

**Anmerkungen:**

Das Programm benötigt eine mitlaufende Uhr. Sie wird innerhalb des Unterprogramms DELAY realisiert, indem die 2-Byte-Variable ZEIT ab Programmstart jede Zehntel-Sekunde um eins erhöht wird.

Von den eintreffenden Signalen der Melde-Einrichtungen wird die Uhrzeit gespeichert. Dabei wird folgendes Schema verwendet:

Es liegt noch keine Meldung vor:	Ankunftszeit = 0,
Es liegt die erste Meldung vor:	Ankunftszeit = Uhrzeit,
Es liegen weitere Meldungen vor:	Keine weitere Speicherung.

Mit Hilfe der Uhr wird auch das Ende der Gelb-Phase gespeichert.

Zu Programmstart ist die Verkehrsdichte noch nicht definiert. Wenn alle Variablen mit null initialisiert werden (dies geschieht z.B. beim Laden des Programms von der Diskette), wird für jeden Verkehrsweg im ersten Durchlauf die Grün-Phasendauer wie bei hoher Verkehrsdichte gewählt.

**Hinweise:**

Der Quelltext des hier abgedruckten Programms ist in zwei Abschnitten auf der Mini-DOS-Diskette gespeichert, damit er mit dem BFZ-Editor selbst dann bearbeitet werden kann, wenn nur 8 KByte Textpuffer zur Verfügung stehen. Die Teile tragen die Dateinamen A10B1.ASM und A10B2.ASM. Die CP/M-Diskette enthält dagegen die Datei an einem Stück (Dateiname A10B.ASM), da der MFA-Mikrocomputer in der CP/M-Version mit 64 KByte RAM ausgestattet ist.

Das Programm kann auch von der Diskette geladen werden (Dateien A10B1.ASM + A10B2.ASM bzw. Datei A10B.MAT), da es vom Kursteilnehmer nicht vervollständigt werden muß. Damit wird Zeit gewonnen, um die beschriebenen Tests sorgfältig durchzuführen.

**Aufgabe A10 (Assembler)**  
**(Blatt 2 von 7)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

```
E000 2A 8CE1    PHASE1: LHL    AN1      ;ANKUNFTZEIT WARTESCHLANGE 1
E003 7C         MOV      A,H      ;TESTE AUF NULL
E004 B5         ORA      L         ;KEINE ANKUNFT = 0
E005 CA 5BE0    JZ       PHASE4    ;WEITER
E008 01 1E00    LXI      B,ZW     ;ROT-GELB-DAUER SIGNAL 1
E00B 3E 8B      MVI      A,8B     ;ROT1 + GELB1 + ROT3
E00D D3 00      OUT      AUSGABE
E00F CD 14E1    CALL     DELAY
E012 01 9600    PHASE2: LXI      B,G1 ;GRUEN-DAUER SIGNAL 1
E015 2A 8EE1    LHL     MARKE1    ;ENDE DER LETZTEN GRUENPHASE
E018 EB        XCHG
E019 2A 8CE1    LHL     AN1      ;ANKUNFTZEIT WARTESCHLANGE 1
E01C CD 7CE1    CALL     SUB2
E01F C2 2BE0    JNZ      AUSG1    ;H-BYTE <> 0
E022 7D        MOV      A,L      ;TESTE L-BYTE
E023 FE 32      CPI      32       ;KRITERIUM VERKEHRSDICHTE
E025 F2 2BE0    JP       AUSG1    ;GERINGE DICHT
E028 CD 83E1    CALL     BMAL2     ;HOHE DICHT
E02B 3E 8C      AUSG1: MVI      A,8C ;GRUEN1 + ROT3
E02D D3 00      OUT      AUSGABE
E02F CD 14E1    CALL     DELAY
E032 01 0100    PHAS2A: LXI      B,0001 ;1/10 SEKUNDE
E035 CD 14E1    CALL     DELAY
E038 2A 90E1    LHL     AN3      ;NUR WEITER, WENN ANFOR-
E03B 7C        MOV      A,H      ;DERUNG VORLIEGT.
E03C B5        ORA      L         ;TESTE AUF NULL
E03D 2A 94E1    LHL     AN5
E040 B4        ORA      H
E041 B5        ORA      L
E042 CA 32E0    JZ       PHAS2A    ;WARTE
E045 01 1E00    PHASE3: LXI      B,ZW ;GELB-DAUER SIGNAL 1
E048 3E 8A      MVI      A,8A     ;GELB1 + ROT3
E04A D3 00      OUT      AUSGABE
E04C CD 14E1    CALL     DELAY
E04F 21 0000    LXI      H,0000   ;REGISTER H,L = 0
E052 22 8CE1    SHLD     AN1      ;ANKUNFTZEIT 1 ZURUECKSETZEN
E055 2A 98E1    LHL     ZEIT     ;ENDE GELBPHASE MERKEN
E058 22 8EE1    SHLD     MARKE1
E05B 2A 90E1    PHASE4: LHL     AN3 ;ANKUNFTZEIT WARTESCHLANGE 3
E05E 7C        MOV      A,H      ;TESTE AUF NULL
E05F B5        ORA      L         ;KEINE ANKUNFT = 0
E060 CA B6E0    JZ       PHASE7    ;WEITER
E063 01 1E00    LXI      B,ZW     ;ROT-GELB-DAUER SIGNAL 3
E066 3E 99      MVI      A,99     ;ROT1 + ROT3 + GELB3
E068 D3 00      OUT      AUSGABE
E06A CD 14E1    CALL     DELAY
E06D 01 6400    PHASE5: LXI      B,G3 ;GRUEN-DAUER SIGNAL 3
E070 2A 92E1    LHL     MARKE3    ;ENDE DER LETZTEN GRUENPHASE
E073 EB        XCHG
E074 2A 90E1    LHL     AN3      ;ANKUNFTZEIT WARTESCHLANGE 1
E077 CD 7CE1    CALL     SUB2
```

---

#### 4. Aufgaben

---

**Aufgabe A10 (Assembler)**  
**(Blatt 3 von 7)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

```
E07A C2 86E0          JNZ     AUSG3    ;H-BYTE <> 0
E07D 7D              MOV     A,L      ;TESTE L-BYTE
E07E FE 32          CPI     32       ;KRITERIUM VERKEHRSDICHTE
E080 F2 86E0          JP      AUSG3    ;GERINGE DICHTe
E083 CD 83E1          CALL    BMAL2    ;HOHE DICHTe
E086 3E A1          AUSG3: MVI     A,0A1  ;ROT1 + GRUEN3
E088 D3 00          OUT     AUSGABE
E08A CD 14E1          CALL    DELAY
E08D 01 0100        PHAS5A: LXI     B,0001 ;1/10 SEKUNDE
E090 CD 14E1          CALL    DELAY
E093 2A 8CE1          LHLD    AN1      ;NUR WEITER, WENN ANFOR-
E096 7C              MOV     A,H      ;DERUNG VORLIEGT.
E097 B5              ORA     L        ;TESTE AUF NULL
E098 2A 94E1          LHLD    AN5
E09B B4              ORA     H
E09C B5              ORA     L
E09D CA 8DE0          JZ      PHAS5A    ;WARTE
EOA0 01 1E00        PHASE6: LXI     B,ZW   ;GELB-DAUER SIGNAL 3
EOA3 3E 91          MVI     A,91      ;ROT1 + GELB3
EOA5 D3 00          OUT     AUSGABE
EOA7 CD 14E1          CALL    DELAY
EOAA 21 0000          LXI     H,0000   ;REGISTER H,L = 0
EOAD 22 90E1          SHLD    AN3      ;ANKUNFTZEIT 3 ZURUECKSETZEN
EOB0 2A 98E1          LHLD    ZEIT     ;ENDE GELBPHASE MERKEN
EOB3 22 92E1          SHLD    MARKE3
EOB6                ;
EOB6                END

0000                ; AUFGABE A10B, TEIL 2
0000                ; STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
0000                ; ENTSPRECHEND DER VERKEHRSFLUSSDICHTe
0000                ;
0000                ; KONSTANTEN:
0000                G1      EQU     0096    ;GRUEN-PHASE 1 = 15 S
0000                G3      EQU     0064    ;GRUEN-PHASE 3 = 10 S
0000                G5      EQU     0064    ;GRUEN-PHASE 5 = 10 S
0000                ZW      EQU     001E    ;UEBERGANGSPHASEN = 3 S
0000                ZEITK   EQU     0A25    ;ZEITKONSTANTE 1/10 S
0000                ;
0000                ; DATENBEREICH:
0000                TASTEN  EQU     0E18B    ;TASTENBYTE
0000                AN1     EQU     0E18C    ;ANKUNFTZEIT 1
0000                MARKE1  EQU     0E18E    ;ENDE LETZTE GRUENPHASE 1
0000                AN3     EQU     0E190    ;ANKUNFTZEIT 3
0000                MARKE3  EQU     0E192    ;ENDE LETZTE GRUENPHASE 3
0000                AN5     EQU     0E194    ;ANKUNFTZEIT 5
0000                MARKE5  EQU     0E196    ;ENDE LETZTE GRUENPHASE 5
0000                ZEIT    EQU     0E198    ;UHRZEIT
0000                ;
```



---

## 4. Aufgaben

---

**Aufgabe A10 (Assembler)**  
(Blatt 4 von 7)

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

```

0000      ; EIN- UND AUSGABE:
0000      EINGABE EQU      0      ; 8-BIT-PARALLEL-EINGABE
0000      AUSGABE EQU      0      ; 8-BIT-PARALLEL-AUSGABE
0000      ;
0000      ; ADRESSEN AUS TEIL 1
0000      PHASE1 EQU      0E000
0000      ;
0000      ORG      0E0B6      ; STARTADRESSE
E0B6      ;
E0B6 2A 94E1      PHASE7: LHLD      AN5      ; ANKUNFTZEIT WARTESCHLANGE 5
E0B9 7C      MOV      A,H      ; TESTE AUF NULL
E0BA B5      ORA      L      ; KEINE ANKUNFT = 0
E0BB CA 00E0      JZ      PHASE1      ; WEITER
E0BE 01 1E00      LXI      B,ZW      ; PAUSEDAUER SIGNAL 5
E0C1 3E 89      MVI      A,89      ; ROT1 + ROT3
E0C3 D3 00      OUT      AUSGABE
E0C5 CD 14E1      CALL     DELAY
E0C8 01 6400      PHASE8: LXI      B,G5      ; GRUEN-DAUER SIGNAL 5
E0CB 2A 96E1      LHLD      MARKE5      ; ENDE DER LETZTEN GRUENPHASE
E0CE EB      XCHG
E0CF 2A 94E1      LHLD      AN5      ; ANKUNFTZEIT WARTESCHLANGE 1
E0D2 CD 7CE1      CALL     SUB2
E0D5 C2 E1E0      JNZ      AUSG5      ; H-BYTE <> 0
E0D8 7D      MOV      A,L      ; TESTE L-BYTE
E0D9 FE 32      CPI      32      ; KRITERIUM VERKEHRSDICHTE
E0DB F2 E1E0      JP      AUSG5      ; GERINGE DICHT
E0DE CD 83E1      CALL     BMAL2      ; HOHE DICHT
E0E1 3E C9      AUSG5: MVI      A,0C9      ; ROT1 + ROT3 + GRUEN5
E0E3 D3 00      OUT      AUSGABE
E0E5 CD 14E1      CALL     DELAY
E0E8 01 0100      PHAS8A: LXI      B,0001      ; 1/10 SEKUNDE
E0EB CD 14E1      CALL     DELAY
E0EE 2A 8CE1      LHLD      AN1      ; NUR WEITER, WENN ANFOR-
E0F1 7C      MOV      A,H      ; DERUNG VORLIEGT.
E0F2 B5      ORA      L      ; TESTE AUF NULL
E0F3 2A 90E1      LHLD      AN3
E0F6 B4      ORA      H
E0F7 B5      ORA      L
E0F8 CA E8E0      JZ      PHAS8A      ; WART
E0FB 01 1E00      PHASE9: LXI      B,ZW      ; PAUSEDAUER SIGNAL 5
E0FE 3E 89      MVI      A,89      ; ROT1 + ROT3
E100 D3 00      OUT      AUSGABE
E102 CD 14E1      CALL     DELAY
E105 21 0000      LXI      H,0000      ; REGISTER H,L = 0
E108 22 94E1      SHLD      AN5      ; ANKUNFTZEIT 5 ZURUECKSETZEN
E10B 2A 98E1      LHLD      ZEIT      ; ENDE GELBPHASE MERKEN
E10E 22 96E1      SHLD      MARKE5
E111      ;
E111 C3 00E0      JMP      PHASE1      ; ENDLOSSCHLEIFE
E114      ;

```

---

#### **4. Aufgaben**

---

**Aufgabe A10 (Assembler)**  
**(Blatt 5 von 7)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

```
E114          ;      UNTERPROGRAMM DELAY
E114 C5      DELAY:  PUSH    B      ;REGISTER RETTEN
E115 D5              PUSH    D
E116 F5              PUSH    PSW
E117 11 250A      DELAY1: LXI    D,ZEITK ;ZEITBASIS 1/10 S
E11A DB 00      DELAY2:  IN     EINGABE ;TASTENSTELLUNGEN EINLESEN
E11C F5              PUSH    PSW      ;KOPIE AUF STACK
E11D 21 8BE1      LXI    H,TASTEN ;SPEICHER FUER TASTENBYTE
E120 AE              XRA    M      ;PRUEFE AUF FLANKENWECHSEL
E121 CA 64E1      JZ     DELAY3 ;NICHTS ZU TUN
E124 F1              POP     PSW
E125 77              MOV     M,A    ;TASTENBYTE SPEICHERN
E126 F5              PUSH    PSW      ;KOPIE AUF STACK
E127 E6 01      TEST1: ANI    01    ;PRUEFE TASTE 1
E129 CA 3AE1      JZ     TEST3    ;NICHTS ZU TUN
E12C 2A 8CE1      LHLD    AN1    ;ANKUNFT 1 BEREITS GESETZT?
E12F 7C              MOV     A,H
E130 B5              ORA     L
E131 C2 3AE1      JNZ     TEST3    ;BEREITS GESETZT!
E134 2A 98E1      LHLD    ZEIT    ;UHRZEIT HOLEN
E137 22 8CE1      SHLD    AN1    ;UND SPEICHERN
E13A F1      TEST3:  POP     PSW    ;TASTENBYTE HOLEN
E13B F5              PUSH    PSW    ;UND WIEDER SICHERN
E13C E6 02      ANI    02    ;PRUEFE TASTE 3
E13E CA 4FE1      JZ     TEST5    ;NICHTS ZU TUN
E141 2A 90E1      LHLD    AN3    ;ANKUNFT 3 BEREITS GESETZT?
E144 7C              MOV     A,H
E145 B5              ORA     L
E146 C2 4FE1      JNZ     TEST5    ;BEREITS GESETZT!
E149 2A 98E1      LHLD    ZEIT    ;UHRZEIT HOLEN
E14C 22 90E1      SHLD    AN3    ;UND SPEICHERN
E14F F1      TEST5:  POP     PSW    ;TASTENBYTE HOLEN
E150 F5              PUSH    PSW    ;UND WIEDER SICHERN
E151 E6 04      ANI    04    ;PRUEFE TASTE 5
E153 CA 64E1      JZ     DELAY3    ;NICHTS ZU TUN
E156 2A 94E1      LHLD    AN5    ;ANKUNFT 5 BEREITS GESETZT?
E159 7C              MOV     A,H
E15A B5              ORA     L
E15B C2 64E1      JNZ     DELAY3    ;BEREITS GESETZT!
E15E 2A 98E1      LHLD    ZEIT    ;UHRZEIT HOLEN
E161 22 94E1      SHLD    AN5    ;UND SPEICHERN
E164 F1      DELAY3: POP     PSW    ;STACK BEREINIGEN
E165 1B              DCX     D      ;GEGEN NULL ZAEHLEN
E166 7A              MOV     A,D    ;TESTE AUF NULL
E167 B3              ORA     E
E168 C2 1AE1      JNZ     DELAY2    ;SCHLEIFENENDE DELAY2
E16B 2A 98E1      LHLD    ZEIT    ;UHRZEIT HOLEN
E16E 23              INX     H      ;UM 1 ERHOEHEN
E16F 22 98E1      SHLD    ZEIT    ;UND SPEICHERN
E172 0B              DCX     B      ;GEGEN NULL ZAEHLEN
E173 78              MOV     A,B    ;AUF NULL TESTEN
```

**Hinweis:**

Nebenstehende Unterprogramme SUB2 und BMAL2 stellen erweiterte Rechenoperationen dar. Insbesondere bei der Programmierung umfangreicher Projekte ist es üblich, sich zunächst ein Paket mit universell einsetzbaren Unterprogrammen zu schaffen.

Solche Unterprogrammsammlungen können thematisch aufgeteilt in Dateien abgespeichert werden, die man üblicherweise als Bibliotheksdateien bezeichnet. Benötigt man die Unterprogramme, so kann man sie aus der Bibliothek herauskopieren oder einfach die komplette Bibliothek dem Programm hinzufügen.

Im vorliegenden Fall könnten die beiden Unterprogramme die Keimzelle einer Arithmetik-Bibliothek sein.

---

**Aufgabe A10 (Assembler)**  
**(Blatt 6 von 7)**

---

**Name:**

---

**Kurs:**

---

**Datum:**

---

```
E174 B1          ORA      C
E175 C2 17E1      JNZ      DELAY1  ;SCHLEIFENENDE DELAY1
E178 F1          POP      PSW      ;REGISTER WIEDER HERSTELLEN
E179 D1          POP      D
E17A C1          POP      B
E17B C9          RET
E17C              ;
E17C              ; UNTERPROGRAMM SUB2
E17C              ; (HL) := (HL) - (DE)
E17C 7D          SUB2:    MOV      A,L
E17D 93          SUB      E
E17E 6F          MOV      L,A
E17F 7C          MOV      A,H
E180 9A          SBB      D
E181 67          MOV      H,A
E182 C9          RET
E183              ;
E183              ; UNTERPROGRAMM BMAL2
E183              ; (BC) := (BC) * 2
E183 79          BMAL2:   MOV      A,C
E184 B7          ORA      A          ;CARRY-BIT LOESCHEN
E185 17          RAL
E186 4F          MOV      C,A
E187 78          MOV      A,B
E188 17          RAL
E189 47          MOV      B,A
E18A C9          RET
E18B              ;
```

Geben Sie das Programm mit Hilfe des Memory- oder Assembler-Kommandos des MAT 85 ein oder laden Sie es von der Diskette.

Prüfen Sie die korrekte Eingabe mit Hilfe des Print- oder Disassembler-Kommandos.

Starten Sie das Programm mit dem GO-Kommando; Startadresse ist E008!

---

#### **4. Aufgaben**

---

---

**Aufgabe A10 (Assembler)**  
**(Blatt 7 von 7)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

Prüfen Sie die korrekte Funktion des Programms. Legen Sie sich selbst die geeigneten Prüfmethoden zurecht und beschreiben Sie sie für die Verkehrsampel 1 in Stichpunkten. Im einzelnen sollten Sie prüfen:

**A10b:** Einhaltung der Zeiten und der korrekten Ampelschaltungen bei normaler Verkehrsdichte.

**A10c:** Verdopplung der Minstdauer der Grün-Phase bei hohem Verkehrsaufkommen.

**A10d:** Korrekte Reaktion auf kein Verkehrsaufkommen bei Verkehrsampel 3 oder Fußgängerampel 5 oder beiden.

**A10e:** Berücksichtigung der Ampeln im Uhrzeigersinn.

▲ **A10f:** Testen Sie weitere selbstgewählte Situationen.

(Beschreibung nachstehend oder auf gesondertem Blatt)

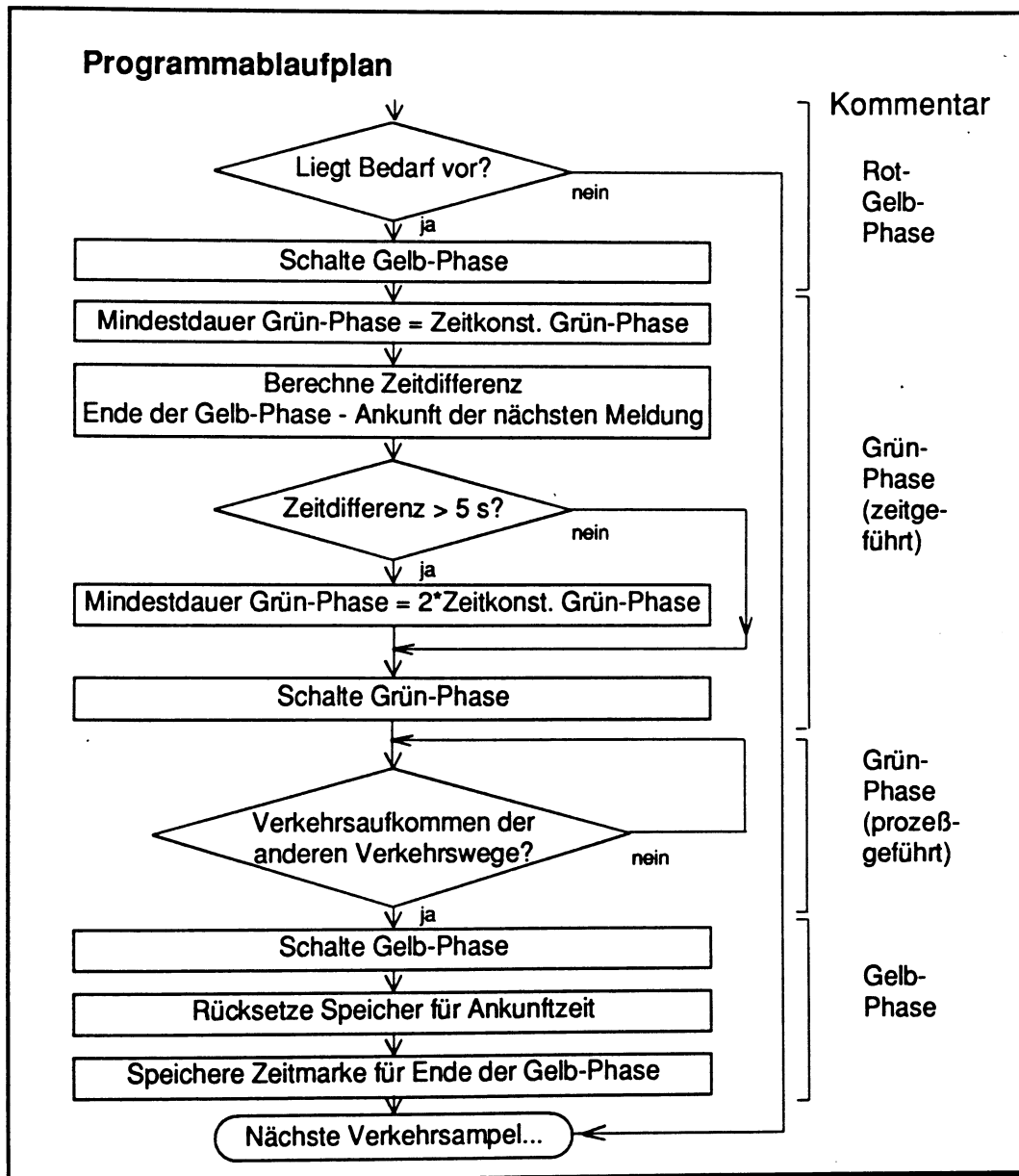
▲ **A10g:** Wann läuft die Uhr zum ersten Mal über, und wie wäre dem abzu-  
helfen?



## 4.10.2 Steuerungsprogramm mit Berücksichtigung der Verkehrsdichte (BASIC)

Im Gegensatz zu den Aufgaben A7, A8 und A9 ist die Behandlung der drei Verkehrswege im folgenden Steuerungsprogramm vollkommen gleich - lediglich die eingestellten Zeitkonstanten geben der Hauptstraße einen Vorzug.

Es genügt daher, den Programmablaufplan für die Steuerung einer Ampel darzustellen. Die beiden anderen Ampeln schließen sich in gleicher Form an, und am Ende steht ein Sprung an den Anfang, um den ganzen Vorgang zyklisch zu wiederholen.



---

**Aufgabe A10 (BASIC)**  
**(Blatt 1 von 3)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

Das nachstehende BASIC-Programm führt die Steuerung der Ampeln in Abhängigkeit von der Verkehrsdichte durch.

```
10 REM AUFGABE A10B
20 REM STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
30 REM ENTSPRECHEND DER VERKEHRSFLUSSDICHTEN
31 REM
32 REM STRATEGIE:
33 REM WENN INNERHALB EINES KOMPLETTEN ZYKLUS DIE
34 REM BEDARFSTASTE BZW. INDUKTIONSSCHLEIFE NICHT
35 REM AKTIVIERT WURDE, WERDEN DIE PHASEN FÜR DIESEN
36 REM VERKEHRSWEG AUSGELASSEN.
37 REM WENN INNERHALB VON 5 SEKUNDEN NACH DER GRÜN-
38 REM PHASE DIE BEDARFSTASTE BZW. INDUKTIONSSCHLEIFE
39 REM AKTIVIERT WURDE, LIEGT EINE HOHE VERKEHRSDICHTE
40 REM VOR. DIE DAUER DER GRÜNPHASE DIESES VERKEHRSWEGS
41 REM WIRD VERDOPPELT.
42 REM
43 D=150 :REM GRÜNPHASE 1 = 15 S
44 F=100 :REM GRÜNPHASE 3 = 10 S
45 G=100 :REM GRÜNPHASE 5 = 10 S
46 H=30 :REM ÜBERGANGSPHASE= 3 S
47 A=0 :REM 8-BIT-PARALLEL-AUSGABE
48 E=0 :REM 8-BIT-PARALLEL-EINGABE
49 Y=0 :REM STARTZEIT = 0
50 K=0 :REM ANKUNFTZEIT 1
51 L=0 :REM ENDE GRÜNPHASE 1
52 M=0 :REM ANKUNFTZEIT 3
53 N=0 :REM ENDE GRÜNPHASE 3
54 O=0 :REM ANKUNFTZEIT 5
55 P=0 :REM ENDE GRÜNPHASE 5
56 GOTO 110 :REM PROGRAMMSTART MIT WEG 1
57 IF K=0 THEN GOTO 400 :REM BEDARF VORHANDEN?
58 Z=H :REM ÜBERGANGSPHASE
59 OUT A,139 :REM ROT1 + GELB1 + ROT3
60 GOSUB 1000
61 Z=D :REM GRÜNPHASE 1
62 IF K<L+50 THEN Z=2*D :REM VERDOPPELN?
63 OUT A,140 :REM GRÜN1 + ROT3
64 GOSUB 1000
65 Z=1 :REM 1/10 SEKUNDE
66 GOSUB 1000
67 IF (M OR O)=0 THEN GOTO 250 :REM BEDARF VORHANDEN?
68 Z=H :REM ÜBERGANGSPHASE
69 OUT A,138 :REM GELB1 + ROT3
70 GOSUB 1000
71 K=0 :REM ANKUNFTZEIT 1 ZURÜCKSETZEN
72 L=Y :REM ENDE GELBPHASE MERKEN
73 IF M=0 THEN GOTO 700 :REM BEDARF VORHANDEN?
74 Z=H :REM ÜBERGANGSPHASE
75 OUT A,153 :REM ROT1 + ROT3 + GELB3
```

**Anmerkungen:**

Das Programm benötigt eine mitlaufende Uhr. Sie wird innerhalb des Unterprogramms VERZOEGERUNG realisiert, indem die Variable Y ab Programmstart jede Zehntel-Sekunde um eins erhöht wird.

Von den eintreffenden Signalen der Melde-Einrichtungen wird die Uhrzeit gespeichert. Dabei wird folgendes Schema verwendet:

Es liegt noch keine Meldung vor:	Ankunftszeit = 0,
Es liegt die erste Meldung vor:	Ankunftszeit = Uhrzeit,
Es liegen weitere Meldungen vor:	Keine weitere Speicherung.

Mit Hilfe der Uhr wird auch das Ende der Gelb-Phase gespeichert.

Das Programm weist einen Nachteil auf:

Der Zeittakt von 1/10 Sekunde kann im Unterprogramm VERZOEGERUNG nicht mehr gewahrt bleiben. Die Anzahl der Durchläufe durch die innerste Schleife muß entweder mit 1 (entsprechend einer Taktzeit von ca. 1/15 Sekunden) oder mit 2 (entsprechend einer Taktzeit von ca. 1/7 Sekunde) festgesetzt werden. Im Programmbeispiel wurden zwei Schleifendurchläufe gewählt. Dadurch fallen alle Schaltzeiten entsprechend länger aus; dies ist bei den Zeitmessungen zu berücksichtigen.

Falls die Programmierung unter CP/M mit MBASIC erfolgt, tritt dieser Nachteil nicht auf. Da MBASIC schneller als Steuer-BASIC arbeitet, kann die innere Schleife achtmal durchlaufen werden. Der Abgleich auf 1/10 Sekunde ist gerade noch mit vertretbarer Genauigkeit möglich.

An dieser Stelle zeigt sich, daß die Grenze der Eignung von BASIC für schnelle Schaltvorgänge erreicht wurde.

**Hinweise:**

Das Programm kann auch von der Diskette geladen werden, da es vom Kursteilnehmer nicht vervollständigt werden muß. Damit wird Zeit gewonnen, um die beschriebenen Tests sorgfältig durchzuführen.

Zu Programmstart ist die Verkehrsdichte noch nicht definiert. Wenn alle Variablen mit null initialisiert werden, wie dies in den Zeilen 80 bis 86 geschieht, wird im ersten Durchlauf für jeden Verkehrsweg die Grün-Phasendauer wie bei hoher Verkehrsdichte gewählt.

---

**Aufgabe A10 (BASIC)**  
**(Blatt 2 von 3)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

```
430 GOSUB 1000
500 Z=F :REM GRUENPHASE 3
510 IF M<N+50 THEN Z=2*F :REM VERDOPPELN?
520 OUT A,161 :REM ROT1 + GRUEN3
530 GOSUB 1000
550 Z=1 :REM 1/10 SEKUNDE
560 GOSUB 1000
570 IF (K OR O)=0 THEN GOTO 550 :REM BEDARF VORHANDEN?
600 Z=H :REM UEBERGANGSPHASE
610 OUT A,145 :REM ROT1 + GELB3
620 GOSUB 1000
630 M=0 :REM ANKUNFTZEIT 3 ZURUECKSETZEN
640 N=Y :REM ENDE GELBPHASE MERKEN
700 IF O=0 THEN GOTO 100 :REM BEDARF VORHANDEN
710 Z=H :REM UEBERGANGSPHASE
720 OUT A,137 :REM ROT1 + ROT3
730 GOSUB 1000
800 Z=G :REM GRUENPHASE 5
810 IF O<P+50 THEN Z=2*G :REM VERDOPPELN?
820 OUT A,201 :REM ROT1 + ROT3 + GRUEN5
830 GOSUB 1000
850 Z=1 :REM 1/10 SEKUNDE
860 GOSUB 1000
870 IF (K OR M)=0 THEN GOTO 850 :REM BEDARF VORHANDEN?
900 Z=H :REM UEBERGANGSPHASE
910 OUT A,137 :REM ROT1 + ROT3
920 GOSUB 1000
930 O=0 :REM ANKUNFTZEIT 5 ZURUECKSETZEN
940 P=Y :REM ENDE GELBPHASE MERKEN
999 GOTO 100 :REM GESAMTPROZESS WIEDERHOLEN
1000 REM VERZOEGERUNG (Z = ZEIT IN 1/10 SEKUNDEN)
1010 REM UND AUFZEICHNUNG VON ZEITMARKEN
1020 FOR I=1 TO Z
1030 FOR J=1 TO 2
1040 IF INP(E)=T THEN GOTO 1100
1050 T=INP(E)
1060 IF (T AND 1)=1 THEN IF K=0 THEN K=Y
1070 IF (T AND 2)=2 THEN IF M=0 THEN M=Y
1080 IF (T AND 4)=4 THEN IF O=0 THEN O=Y
1100 NEXT J
1110 Y=Y+1
1120 NEXT I
1130 RETURN
```

Geben Sie das Programm ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.

Starten Sie das Programm mit dem Kommando RUN.

---

#### **4. Aufgaben**

---

---

**Aufgabe A10 (BASIC)**  
**(Blatt 3 von 3)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

Prüfen Sie die korrekte Funktion des Programms. Legen Sie sich selbst die geeigneten Prüfmethoden zurecht und beschreiben Sie sie für die Verkehrsampel 1 in Stichpunkten. Im einzelnen sollten Sie prüfen:

**A10b:** Einhaltung der Zeiten (Achtung: die Zeiten sind etwa 1,4 mal so lang!) und der korrekten Ampelschaltungen bei normaler Verkehrsdichte.

**A10c:** Verdopplung der Mindestdauer der Grün-Phase bei hohem Verkehrsaufkommen.

**A10d:** Korrekte Reaktion auf kein Verkehrsaufkommen bei Verkehrsampel 3 oder Fußgängerampel 5 oder beiden.

**A10e:** Berücksichtigung der Ampeln im Uhrzeigersinn.

▲ **A10f:** Testen Sie weitere selbstgewählte Situationen.

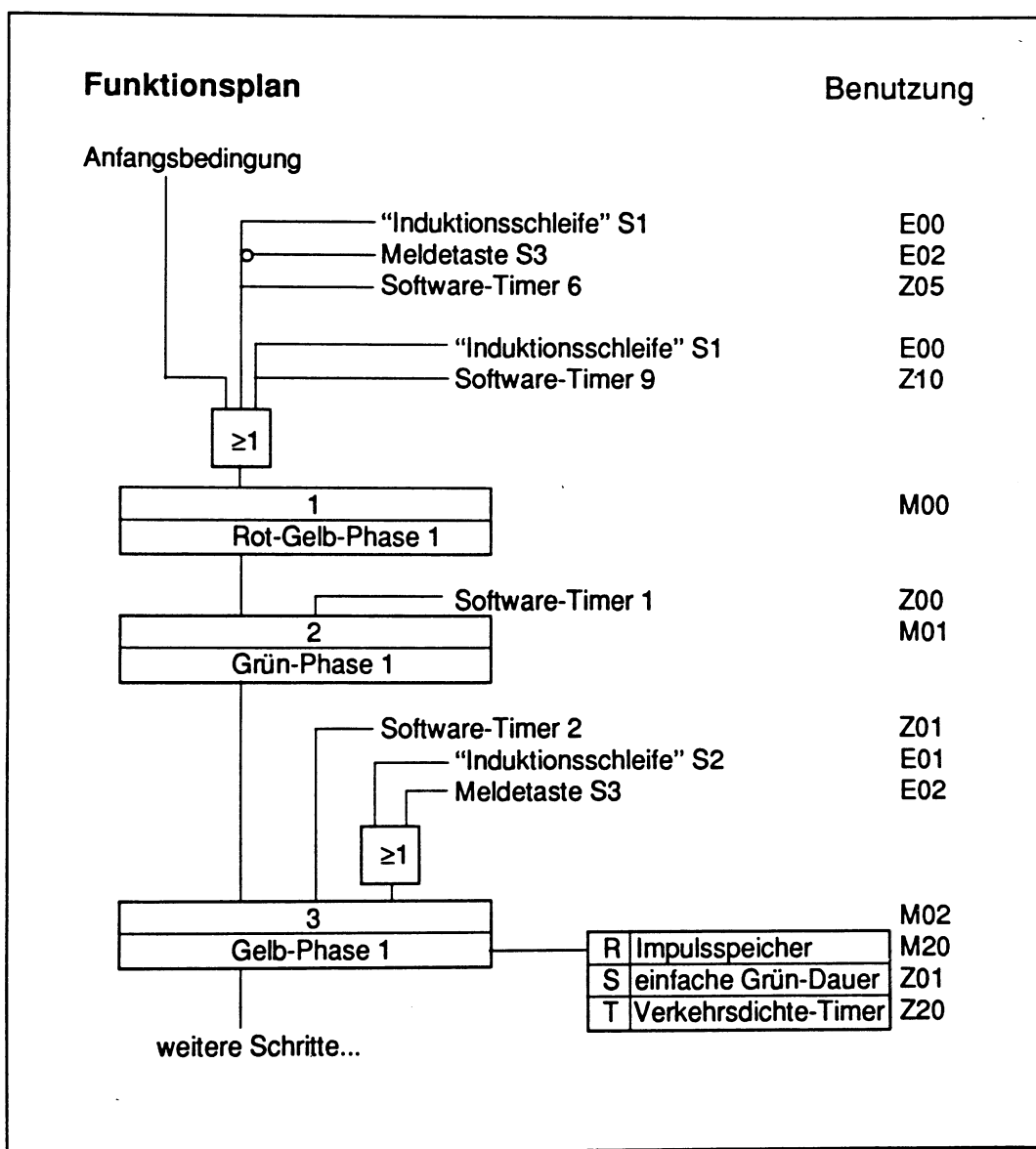
(Beschreibung nachstehend oder auf gesondertem Blatt)

▲ **A10g:** Wann läuft die Uhr zum ersten Mal über, und wie wäre dem abzu-  
helfen?

## 4.10.3 Steuerungsprogramm mit Berücksichtigung der Verkehrsdichte (SPS)

Im Gegensatz zu den Aufgaben A7, A8 und A9 ist die Behandlung der drei Verkehrswege im folgenden Steuerungsprogramm vollkommen gleich - lediglich die eingestellten Zeitkonstanten geben der Hauptstraße einen Vorzug.

Es genügt daher, den Funktionsplan für die Schrittkette nur einer Ampel darzustellen. Die beiden anderen Ampeln schließen sich in gleicher Form an und am Ende wird wieder der erste Schritt aktiviert, um den ganzen Vorgang zyklisch zu wiederholen.



---

**Aufgabe A10 (SPS)**  
**(Blatt 1 von 6)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

Das nachstehende SPS-Programm führt die Steuerung der Ampeln in Abhängigkeit der Verkehrsdichte durch.

```
; AUFGABE A10B
; STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
; ENTSPRECHEND DER VERKEHRSFLUSSDICHTEN

; STRATEGIE:
; WENN INNERHALB EINES KOMPLETTEN ZYKLUS DIE
; BEDARFSTASTE BZW. INDUKTIONSSCHLEIFE NICHT
; AKTIVIERT WURDE, WERDEN DIE PHASEN FÜR DIESEN
; VERKEHRSWEG AUSGELASSEN.
; WENN INNERHALB VON 5 SEKUNDEN NACH DER GELB- BZW.
; PAUSENPHASE DIE BEDARFSTASTE BZW. INDUKTIONSSCHLEIFE
; AKTIVIERT WURDE, LIEGT EINE HOHE VERKEHRSDICHTE VOR.
; DIE DAUER DER GRÜNPHASE DIESES VERKEHRSWEGS WIRD
; IM NÄCHSTEN ZYKLUS VERDOPPELT.
; INITIALISIEREN DER ZEITGEBER
```

```
UN M 00      ; ANFANGSBEDINGUNG:
UN M 01      ; ALLE MERKER NICHT
UN M 02      ; GESETZT.
UN M 03      ;
UN M 04      ;
UN M 05      ;
UN M 06      ;
UN M 07      ;
UN M 10      ;
=L Z 00,30    ;1. ROT-GELB-PHASE AMPEL 1
=L Z 01,150   ;2. GRÜN-PHASE      AMPEL 1
=L Z 02,30    ;3. GELB-PHASE      AMPEL 1
=L Z 03,30    ;4. ROT-GELB-PHASE AMPEL 3
=L Z 04,100   ;5. GRÜN-PHASE      AMPEL 3
=L Z 05,30    ;6. GELB-PHASE      AMPEL 3
=L Z 06,30    ;7. PAUSENPHASE     AMPEL 5
=L Z 07,100   ;8. GRÜNPHASE       AMPEL 5
=L Z 10,30    ;9. PAUSENPHASE     AMPEL 5
=L Z 20,50    ; ZEITINTERVALL VERKEHRSDICHTE
=S A 07      ; FUSSGAENGERAMPELN FREIGEBEN
=S M 00      ;
=S Z 00      ;1  STARTE PHASE 1
```

```
; ABLAUFKETTE (GESTEUERT DURCH ZEITGEBER)
```

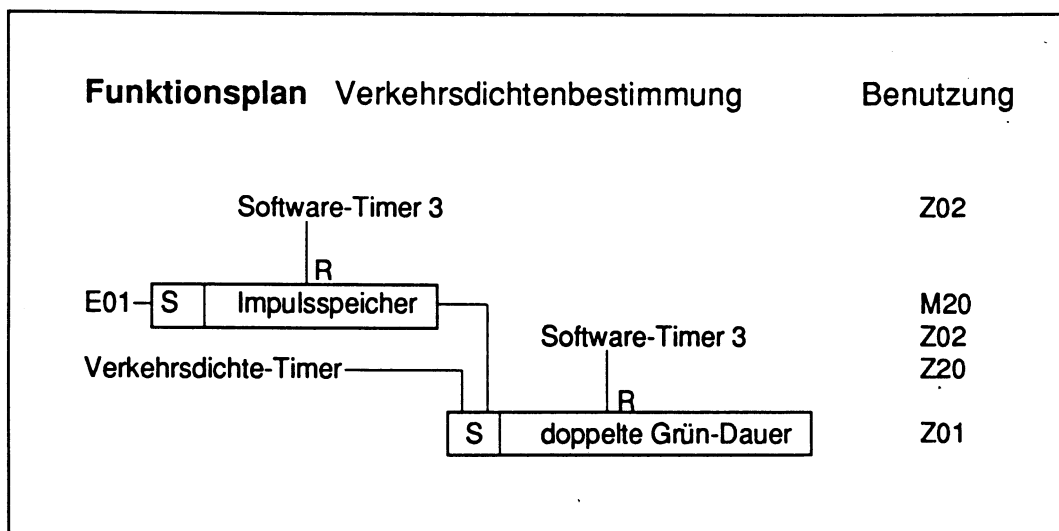
```
U   Z 00      ; PHASE 1 ABGELAUFEN
=R  Z 00      ;
=R  M 00      ;
=S  Z 01      ; STARTE PHASE 2
=S  M 01      ;2
```



In die Schrittsteuerung ist nun - im Vergleich zu den Vorgängerprogrammen - die Bestimmung der Verkehrsdichte aufgenommen. Mit dem Ende der Gelb-Phase wird ein Software-Timer gestartet, der von einem Hilfsmerker begleitet wird, der den Zählzustand des Software-Timers anzeigt.

Alle einlaufenden Signale der Melde-Einrichtungen werden nicht nur wie bisher gespeichert, sondern auch mit obigem Hilfsmerker in Verbindung gebracht. Sind sowohl der Eingang als auch der Hilfsmerker gesetzt, liegt hohe Verkehrsdichte vor, und der Software-Timer für die Grün-Phase wird mit dem doppelten Wert geladen.

Das Zurückschalten des Software-Timers auf die einfache Dauer erfolgt vorsorglich am Ende einer jeden Gelb-Phase.



**Aufgabe A10 (SPS)**  
**(Blatt 2 von 6)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

```
U  M 21      ;   WENN FAHRZEUG AN IND.SCHL. 3
O  M 22      ;   ODER FUSSGAENGER AN TASTE 5
=  M 23      ;3  HILFSMERKER

U  M 23      ;   S.O.
U  Z 01      ;   UND PHASE 2 ABGELAUFEN
=R Z 01      ;
=R M 01      ;
=S Z 02      ;   STARTE PHASE 3
=S M 02      ;4

U  Z 02      ;   WENN PHASE 3 ABGELAUFEN
=R M 20      ;   RUECKSETZE IMPULSSPEICHER DER IND.SCHL. 1
=L Z 01,150  ;   GRUENPHASE AMPEL 1 = 15 SEKUNDEN
=S Z 20      ;   STARTE VERKEHRSDICHTE-TIMER
=S M 30      ;5  MERKER MESSPHASE AMPEL 1

U  M 21      ;   WENN FAHRZEUG AN IND.SCHL. 3
U  Z 02      ;   UND PHASE 3 ABGELAUFEN
=R Z 02      ;
=R M 02      ;
=S Z 03      ;   STARTE PHASE 4
=S M 03      ;6

U  M 22      ;   WENN FUSSGAENGER AN TASTE 5
UN M 21      ;   UND KEIN FAHRZEUG AN IND.SCHL. 3
U  Z 02      ;   UND PHASE 3 ABGELAUFEN
=R Z 02      ;
=R M 02      ;
=S Z 06      ;   STARTE PHASE 7
=S M 06      ;7

U  Z 03      ;   PHASE 4 ABGELAUFEN
=R Z 03      ;
=R M 03      ;
=S Z 04      ;   STARTE PHASE 5
=S M 04      ;8

U  M 20      ;   WENN FAHRZEUG AN IND.SCHL. 1
O  M 22      ;   ODER FUSSGAENGER AN TASTE 5
=  M 24      ;9  HILFSMERKER

U  M 24      ;   S.O.
U  Z 04      ;   UND PHASE 5 ABGELAUFEN
=R Z 04      ;
=R M 04      ;
=S Z 05      ;   STARTE PHASE 6
=S M 05      ;10
```

**Hinweis:**

Wenn der Kursteilnehmer den BFZ-SPS-Interpreter in der Version der FTÜ benutzt, so wird von nachstehendem gleichwertigen Programm ausgegangen.

```
*/M00*/M01*/M02*/M03*/M04*/M05*/M06*/M07*/M10=LZ00,30=LZ01,150
=LZ02,30=LZ03,30=LZ04,100=LZ05,30=LZ06,30=LZ07,100=LZ10,30=LZ20,50
=SA07=SM00=SZ00
*Z00=RZ00=RM00=SZ01=SM01
*M21+M22=M23
*M23*Z01=RZ01=RM01=SZ02=SM02
*Z02=RM20=LZ01,150=SZ20=SM30
*M21*Z02=RZ02=RM02=SZ03=SM03
*M22*/M21*Z02=RZ02=RM02=SZ06=SM06
*Z03=RZ03=RM03=SZ04=SM04
*M20+M22=M24
*M24*Z04=RZ04=RM04=SZ05=SM05
*Z05=RM21=LZ04,100=SZ20=SM31
*M22*Z05=RZ05=RM05=SZ06=SM06
*M20*/M22*Z05=RZ05=RM05=SZ00=SM00
*Z06=RZ06=RM06=SZ07=SM07
*M20+M21=M25
*M25*Z07=RZ07=RM07=SZ10=SM10
*Z10=RM22=LZ07,100=SZ20=SM32
*M20*Z10=RZ10=RM10=SZ00=SM00
*M21*/M20*Z10=RZ10=RM10=SZ03=SM03
*E00=SM20
*E01=SM21
*E02=SM22
*Z20=RZ20=RM30=RM31=RM32
*M20*M30=LZ01,300
*M21*M31=LZ04,200
*M22*M32=LZ07,200
*M00+M03+M04+M05+M06+M07+M10=A00
*M00+M02=A01
*M01=A02
*M00+M01+M02+M03+M06+M07+M10=A03
*M03+M05=A04
*M04=A05
*M07=A06
```

**Aufgabe A10 (SPS)**  
**(Blatt 3 von 6)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

```
U   Z 05      ;   WENN PHASE 6 ABGELAUFEN
=R  M 21      ;   RUECKSETZE IMPULSSPEICHER DER IND.SCHL. 3
=L  Z 04,100  ;   GRUENPHASE AMPEL 3 = 10 SEKUNDEN
=S  Z 20      ;   STARTE VERKEHRSDICHTE-TIMER
=S  M 31      ;11 MERKER MESSPHASE AMPEL 3

U   M 22      ;   WENN FUSSGAENGER AN TASTE 5
U   Z 05      ;   UND PHASE 6 ABGELAUFEN
=R  Z 05      ;
=R  M 05      ;
=S  Z 06      ;   UND STARTE PHASE 7
=S  M 06      ;12

U   M 20      ;   WENN FAHRZEUG AN IND.SCHL. 1
UN  M 22      ;   UND KEIN FUSSGAENGER AN TASTE 5
U   Z 05      ;   UND PHASE 6 ABGELAUFEN
=R  Z 05      ;
=R  M 05      ;
=S  Z 00      ;   STARTE PHASE 1
=S  M 00      ;13

U   Z 06      ;   PHASE 7 ABGELAUFEN
=R  Z 06      ;
=R  M 06      ;
=S  Z 07      ;   STARTE PHASE 8
=S  M 07      ;14

U   M 20      ;   WENN FAHRZEUG AN IND.SCHL. 1
O   M 21      ;   ODER FAHRZEUG AN IND.SCHL. 3
=   M 25      ;15 HILFSMERKER

U   M 25      ;   S.O.
U   Z 07      ;   UND PHASE 8 ABGELAUFEN
=R  Z 07      ;
=R  M 07      ;
=S  Z 10      ;   STARTE PHASE 9
=S  M 10      ;16

U   Z 10      ;   WENN PHASE 9 ABGELAUFEN
=R  M 22      ;   RUECKSETZE IMPULSSPEICHER TASTE 5
=L  Z 07,100  ;   GRUENPHASE AMPEL 5 = 10 SEKUNDEN
=S  Z 20      ;   STARTE VERKEHRSDICHTE-TIMER
=S  M 32      ;17 MERKER MESSPHASE AMPEL 5

U   M 20      ;   WENN FAHRZEUG AN IND.SCHL. 1
U   Z 10      ;   UND PHASE 9 ABGELAUFEN
=R  Z 10      ;
=R  M 10      ;
=S  Z 00      ;   STARTE PHASE 1
=S  M 00      ;18
```

**Hinweis:**

Das Programm kann auch von der Diskette geladen werden (Datei A10B.DOK bzw. Datei A10B.DIN bzw. Datei A10B.SPS), da es vom Kursteilnehmer nicht vervollständigt werden muß. Damit wird Zeit gewonnen, um die beschriebenen Tests sorgfältig durchzuführen.

**Aufgabe A10 (SPS)**  
**(Blatt 4 von 6)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

```
U   M 21      ;   WENN FAHRZEUG AN IND.SCHL. 3
UN  M 20      ;   UND KEIN FAHRZEUG AN IND.SCHL. 1
U   Z 10      ;   UND PHASE 9 ABGELAUFEN
=R  Z 10      ;
=R  M 10      ;
=S  Z 03      ;   STARTE PHASE 4
=S  M 03      ;19
```

; SPEICHERUNG DER EINGABEIMPULSE

```
U   E 00      ;   INDUKTIONSSCHLEIFE 1
=S  M 20      ;20
```

```
U   E 01      ;   INDUKTIONSSCHLEIFE 3
=S  M 21      ;21
```

```
U   E 02      ;   FUSSGAENGERTASTE 5
=S  M 22      ;22
```

; VERKEHRSDICHTENSTEUERUNG

```
U   Z 20      ;   WENN VERKEHRSDICHTE-TIMER ABGELAUFEN
=R  Z 20      ;   WIEDER ZURUECKSETZEN
=R  M 30      ;   HILFSMERKER EBENFALLS
=R  M 31      ;   ZURUECKSETZEN
=R  M 32      ;23
```

```
U   M 20      ;   WENN IMPULSSPEICHER IND.SCHL. 1
U   M 30      ;   UND VERKEHRSDICHTE-TIMER
=L  Z 01,300   ;24 GRUENPHASE AMPEL 1 = 30 SEKUNDEN
```

```
U   M 21      ;   WENN IMPULSSPEICHER IND.SCHL. 3
U   M 31      ;   UND VERKEHRSDICHTE-TIMER
=L  Z 04,200   ;25 GRUENPHASE AMPEL 3 = 20 SEKUNDEN
```

```
U   M 22      ;   WENN IMPULSSPEICHER TASTE 5
U   M 32      ;   UND VERKEHRSDICHTE-TIMER
=L  Z 07,200   ;26 GRUENPHASE AMPEL 5 = 20 SEKUNDEN
```

; AUSGABESTEUERUNG

```
U   M 00      ;   IN PHASE 1
O   M 03      ;   UND PHASE 4
O   M 04      ;   UND PHASE 5
O   M 05      ;   UND PHASE 6
O   M 06      ;   UND PHASE 7
O   M 07      ;   UND PHASE 8
O   M 10      ;   UND PHASE 9
=   A 00      ;27 ROT1 SCHALTEN
```

---

#### **4. Aufgaben**

---

---

**Aufgabe A10 (SPS)**  
**(Blatt 5 von 6)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

```
U  M 00      ;    IN  PHASE 1
O  M 02      ;    UND PHASE 3
=  A 01      ;28 GELB1 SCHALTEN

U  M 01      ;    IN  PHASE 2
=  A 02      ;29 GRUEN1 SCHALTEN

U  M 00      ;    IN  PHASE 1
O  M 01      ;    UND PHASE 2
O  M 02      ;    UND PHASE 3
O  M 03      ;    UND PHASE 4
O  M 06      ;    UND PHASE 7
O  M 07      ;    UND PHASE 8
O  M 10      ;    UND PHASE 9
=  A 03      ;30 ROT3  SCHALTEN

U  M 03      ;    IN  PHASE 4
O  M 05      ;    UND PHASE 6
=  A 04      ;31 GELB3 SCHALTEN

U  M 04      ;    IN PHASE 5
=  A 05      ;32 GRUEN3 SCHALTEN

U  M 07      ;    IN PHASE 8
=  A 06      ;33 GRUEN5 SCHALTEN

END
```

Geben Sie das Programm ein.

Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.

Starten Sie das Programm mit dem GO-Kommando.



---

#### **4. Aufgaben**

---

---

**Aufgabe A10 (SPS)**  
**(Blatt 6 von 6)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

Prüfen Sie die korrekte Funktion des Programms. Legen Sie sich selbst die geeigneten Prüfmethoden zurecht und beschreiben Sie sie für die Verkehrsampel 1 in Stichpunkten. Im einzelnen sollten Sie prüfen:

**A10b:** Einhaltung der Zeiten und der korrekten Ampelschaltungen bei normaler Verkehrsdichte.

**A10c:** Verdopplung der Mindestdauer der Grün-Phase bei hohem Verkehrsaufkommen.

**A10d:** Korrekte Reaktion auf kein Verkehrsaufkommen bei Verkehrsampel 3 oder Fußgängerampel 5 oder beiden.

**A10e:** Berücksichtigung der Ampeln im Uhrzeigersinn.

▲ **A10f:** Testen Sie weitere selbstgewählte Situationen.

(Beschreibung nachstehend oder auf gesondertem Blatt).

**4.11 Aufgabe 11: Tag/Nacht-Umschaltung**

Die Ampelsteuerung aus Aufgabe A10 soll um eine Nacht-Schaltung ergänzt werden. Hierzu steht eine Schaltuhr (ggf. auch ein Instrument zur Messung der Helligkeit des Umgebungslichts) zur Verfügung. Der Ausgang der Uhr ist an den Eingang E03 angeschlossen (siehe Technologieschema in *Kapitel 2.3*, Beschreibung der Betriebsmittel in *Kapitel 2.7* und die Zuordnungsliste in *Kapitel 3.3*).

Bei der Bearbeitung der Aufgaben wird die Schaltuhr lediglich durch den Schalter an der 8-Bit-Parallel-Eingabe simuliert.

Die Schaltuhr liefert ein Low-Signal (Schalter an der 8-Bit-Parallel-Eingabe in rechter Stellung) während der Tageszeit. Unter diesen Bedingungen läuft die gleiche Steuerung wie in Aufgabe A10 ab.

Die Schaltuhr liefert während der Nachtzeit ein High-Signal (Schalter an der 8-Bit-Parallel-Eingabe in linker Stellung). In dieser Situation werden die Fußgängerampeln abgeschaltet, und die beiden Verkehrsampeln weisen gelbes Blinklicht auf.

Das Blinken der Verkehrsampeln besteht aus einer Hellphase von 0,5 Sekunden Dauer, in der H2 und H7 eingeschaltet sind, und einer Pausenphase von 0,5 Sekunden Dauer, in der alle Lampen ausgeschaltet sind.

Der Übergang zwischen Tagbetrieb und Nachtbetrieb darf nicht zu beliebigen Zeitpunkten erfolgen, um die Verkehrsteilnehmer nicht zu verunsichern. Ein sinnvoller Übergang vom Tag- zum Nachtbetrieb erfolgt dann, wenn die zeitgeführte Grün-Phase des gerade freigegebenen Verkehrswegs beendet ist. Programmtechnisch wird die Abfrage der Schaltuhr in die erweiterte prozeßgeführte Grün-Phase eingebunden.

Der Übergang vom Nachtbetrieb zum Tagbetrieb erfolgt (willkürlich festgelegt) am Ende der Pausenphase und startet mit der Freigabe der Hauptstraße (wie bei Programmstart).

Während des Nachtbetriebs können alle Verkehrsteilnehmer unter Beachtung der üblichen Verkehrsregeln den Einmündungsbereich zu beliebigen Zeitpunkten passieren. Signale der Melde-Einrichtungen müssen daher nicht gespeichert werden, bzw. bereits im Tagbetrieb gespeicherte und noch nicht abgearbeitete Meldungen können verworfen werden. Programmtechnisch geschieht dies durch das Löschen der betreffenden Speicher beim Übergang vom Nacht- zum Tagbetrieb.

**Hinweis:**

Fortsetzung der Aufgabe A11

- für Assembler-Programmierung auf S. 178,
- für BASIC-Programmierung auf S. 194,
- für SPS-Programmierung auf S. 200.

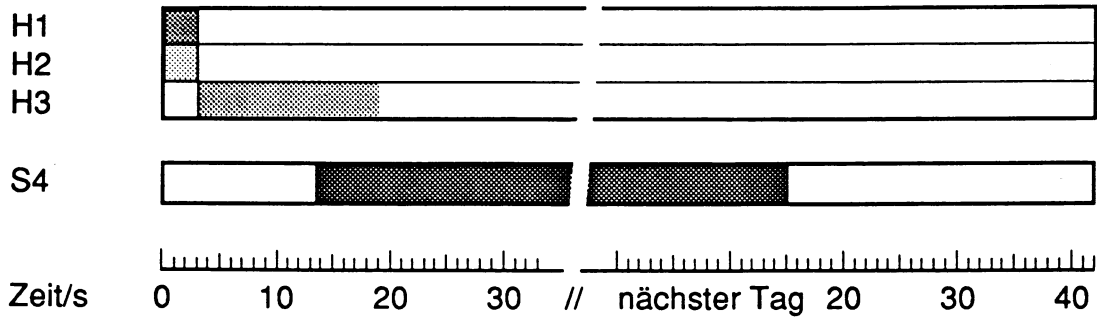
## Aufgabe A11 (Einführung)

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

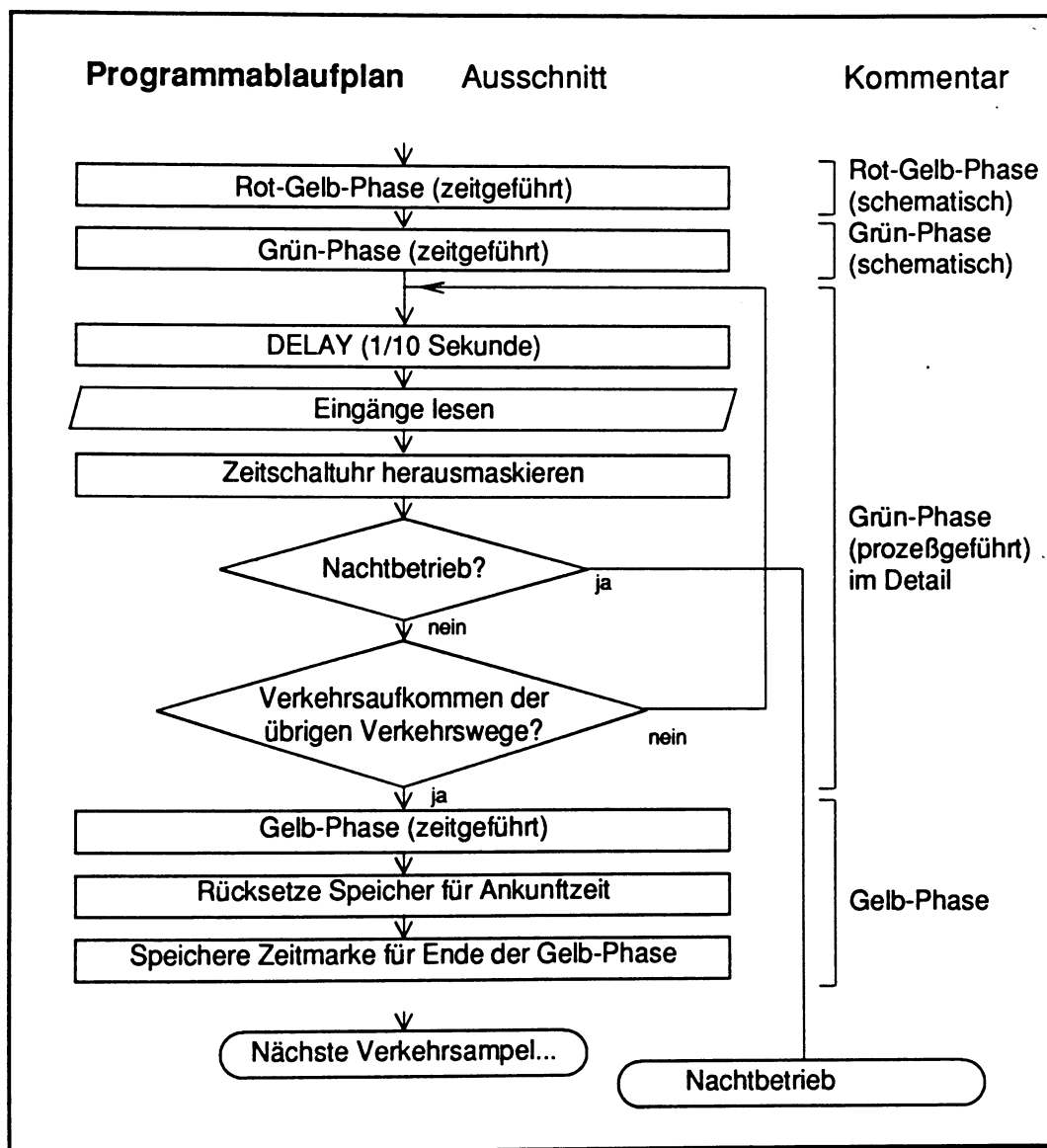
Datum: \_\_\_\_\_

**A11a:** Ergänzen Sie das nachstehende Signal-Zeit-Diagramm:



## 4.11.1 Betriebsarten-Umschaltung (Assembler)

Die Abfrage der Schaltuhr wird in die Erweiterung der Grün-Phase (prozeßgeführter Schritt) eingefügt. Dies geschieht in allen drei Grün-Phasen, denn sonst könnte es vorkommen, daß bei mangelndem Verkehrsaufkommen die Steuerung in einer Grün-Phase ohne Abfrage verweilt. Der nachstehende Programmablaufplan gilt daher für alle drei Verkehrswege gleichermaßen (das Programm behandelt genauso wie jenes aus A10 alle drei Verkehrswege gleichwertig). Der zweite Abschnitt des Programmablaufplans enthält den Nachtbetrieb zusammen mit dem Übergang zum Tagbetrieb.



---

**Aufgabe A11 (Assembler)**  
**(Blatt 1 von 8)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

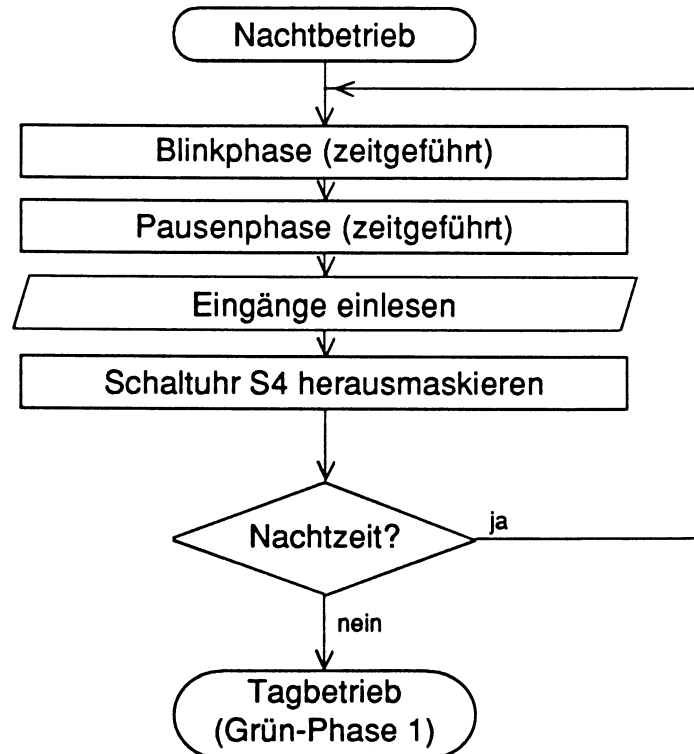
---

**A11b:** Das nachstehende Assemblerprogramm führt die Steuerung der Ampeln sowohl im Tag- als auch im Nachtbetrieb durch.

Das Programm ist eine Weiterentwicklung desjenigen aus Aufgabe A10. Es enthält bereits die korrekten Übergangsbedingungen vom Tag- zum Nachtbetrieb. Der Nachtbetrieb und der Übergang zum Tagbetrieb ist noch nicht programmiert. Füllen Sie die Leerstellen im Programm aus.

```
0000      ; AUFGABE A11B, TEIL 1
0000      ; STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
0000      ; ENTSPRECHEND DER VERKEHRSFLUSSDICHTEN.
0000      ; TAG/NACHT-SCHALTUNG GEMAESS SCHALTUHR S4.
0000      ;
0000      ; STRATEGIE:
0000      ; WENN INNERHALB EINES KOMPLETTEN ZYKLUS DIE
0000      ; BEDARFSTASTE BZW. INDUKTIONSSCHLEIFE NICHT
0000      ; AKTIVIERT WURDE, WERDEN DIE PHASEN FUEER DIESEN
0000      ; VERKEHRSWEG AUSGELASSEN.
0000      ; WENN INNERHALB VON 5 SEKUNDEN NACH DER GRUEN-
0000      ; PHASE DIE BEDARFSTASTE BZW. DIE INDUKTIONSSCHLEIFE
0000      ; AKTIVIERT WURDE, LIEGT EINE HOHE VERKEHRSDICHTE
0000      ; VOR. DIE GRUENPHASE DIESES VERKEHRSWEGS WIRD VER-
0000      ; DOPPELT.
0000      ;
0000      ; KONSTANTEN:
0000      G1      EQU      0096      ;GRUEN-PHASE 1 = 15 S
0000      G3      EQU      0064      ;GRUEN-PHASE 3 = 10 S
0000      G5      EQU      0064      ;GRUEN-PHASE 5 = 10 S
0000      ZW      EQU      001E      ;UEBERGANGSPHASEN = 3 S
0000      ZEITK   EQU      0A25      ;ZEITKONSTANTE 1/10 S
0000      ;
0000      ; DATENBEREICH:
0000      TASTEN  EQU      0E1DD      ;TASTENBYTE
0000      AN1     EQU      0E1DE      ;ANKUNFTZEIT 1
0000      MARKE1  EQU      0E1E0      ;ENDE LETZTE GRUENPHASE 1
0000      AN3     EQU      0E1E2      ;ANKUNFTZEIT 3
0000      MARKE3  EQU      0E1E4      ;ENDE LETZTE GRUENPHASE 3
0000      AN5     EQU      0E1E6      ;ANKUNFTZEIT 5
0000      MARKE5  EQU      0E1E8      ;ENDE LETZTE GRUENPHASE 5
0000      ZEIT    EQU      0E1EA      ;UHRZEIT
0000      ;
0000      ; ADRESSEN AUS TEIL 2
0000      PHASE7  EQU      0E0CA
0000      PHAS10  EQU      0E12F
0000      DELAY   EQU      0E150      ;VERZOEGERUNGSRoutine
0000      SUB2    EQU      0E1B8      ;16-BIT-SUBTRAKTIONSRoutine
0000      BMAL2   EQU      0E1BF      ;16-BIT-LINKSSCHIEBEN
0000      INIT    EQU      0E1C7      ;DATENBEREICH LOESCHEN
0000      ;
```

## Programmablaufplan



**Aufgabe A11 (Assembler)**  
**(Blatt 2 von 8)**

Name:

Kurs:

Datum:

```
0000      ; EIN- UND AUSGABE:
0000      EINGABE EQU    0      ;8-BIT-PARALLEL-EINGABE
0000      AUSGABE EQU    0      ;8-BIT-PARALLEL-AUSGABE
0000      ;
0000      ORG    0E000    ;STARTADRESSE
E000      ;
E000 CD C7E1      CALL    INIT    ;DATENBEREICH LOESCHEN
E003 C3 0EE0      JMP     START    ;MIT PHASE 1 ANFANGEN
E006 2A DEE1      PHASE1: LHLD    AN1    ;ANKUNFTZEIT WARTESCHLANGE 1
E009 7C          MOV     A,H    ;TESTE AUF NULL
E00A B5          ORA     L    ;KEINE ANKUNFT = 0
E00B CA 68E0      JZ      PHASE4    ;WEITER
E00E 01 1E00      START: LXI     B,ZW    ;ROT-GELB-DAUER SIGNAL 1
E011 3E 8B      MVI     A,8B    ;ROT1 + GELB1 + ROT3
E013 D3 00      OUT     AUSGABE
E015 CD 50E1      CALL    DELAY
E018 01 9600      PHASE2: LXI     B,G1    ;GRUEN-DAUER SIGNAL 1
E01B 2A E0E1      LHLD    MARKE1    ;ENDE DER LETZTEN GRUENPHASE
E01E EB          XCHG
E01F 2A DEE1      LHLD    AN1    ;ANKUNFTZEIT WARTESCHLANGE 1
E022 CD B8E1      CALL    SUB2
E025 C2 31E0      JNZ     AUSG1    ;H-BYTE <> 0
E028 7D          MOV     A,L    ;TESTE L-BYTE
E029 FE 32      CPI     32    ;KRITERIUM VERKEHRSDICHTE
E02B F2 31E0      JP      AUSG1    ;GERINGE DICHT
E02E CD BFE1      CALL    BMAL2    ;HOHE DICHT
E031 3E 8C      AUSG1: MVI     A,8C    ;GRUEN1 + ROT3
E033 D3 00      OUT     AUSGABE
E035 CD 50E1      CALL    DELAY
E038 01 0100      PHAS2A: LXI     B,0001    ;1/10 SEKUNDE
E03B CD 50E1      CALL    DELAY
E03E      ; UEBERPRUEFEN, OB NACHT-SCHALTUNG
E03E DB 00      IN      EINGABE    ;EINGAENGE LESEN
E040 E6 08      ANI     08    ;SCHALTUHR MASKIEREN
E042 C2 2FE1      JNZ     PHAS10    ;NUN NACHT-BETRIEB
E045      ; UEBERPRUEFEN DER MELDEEINRICHTUNGEN
E045 2A E2E1      LHLD    AN3    ;NUR WEITER, WENN ANFOR-
E048 7C          MOV     A,H    ;DERUNG VORLIEGT.
E049 B5          ORA     L    ;TESTE AUF NULL
E04A 2A E6E1      LHLD    AN5
E04D B4          ORA     H
E04E B5          ORA     L
E04F CA 38E0      JZ      PHAS2A    ;WARTE
E052 01 1E00      PHASE3: LXI     B,ZW    ;GELB-DAUER SIGNAL 1
E055 3E 8A      MVI     A,8A    ;GELB1 + ROT3
E057 D3 00      OUT     AUSGABE
E059 CD 50E1      CALL    DELAY
E05C 21 0000      LXI     H,0000    ;REGISTER H,L = 0
E05F 22 DEE1      SHLD    AN1    ;ANKUNFTZEIT 1 ZURUECKSETZEN
E062 2A EAE1      LHLD    ZEIT    ;ENDE GELBPHASE MERKEN
E065 22 E0E1      SHLD    MARKE1
```



**Hinweise:**

Der Quelltext des Programms ist in zwei Abschnitten auf der Mini-DOS-Diskette gespeichert, damit er mit dem BFZ-Editor selbst dann bearbeitet werden kann, wenn nur 8 KByte Textpuffer zur Verfügung stehen. Die Teile tragen die Dateinamen A11B1.ASM und A11B2.ASM. Die CP/M-Diskette enthält dagegen die Datei an einem Stück (Dateiname A11B.ASM), da der MFA-Mikrocomputer in der CP/M-Version mit 64 KByte RAM ausgestattet ist.

Da das Programm der Aufgabe A11b einen erheblichen Umfang besitzt, kann zum zügigen Bearbeiten der Aufgabenstellung dem Kursteilnehmer eine Programmvariante zur Verfügung gestellt werden, bei der die Leerstellen noch nicht ausgefüllt sind.

Der Assembler-Quelltext besteht in diesem Fall aus den Dateien A11B1.ASM und A11BLEER.ASM (die Leerstellen befinden sich nur im zweiten Teil des Quelltextes).

Bei der Arbeit mit den Maschinenprogrammen mit Hilfe der MAT-85-Kommandos kann entsprechend die Datei A11BLEER.MAT zur Verfügung gestellt werden. Sie enthält dort, wo der Quelltext die Leerstellen aufweist, Null-Bytes.

Zu Programmstart ist die Verkehrsdichte noch nicht definiert. Da im Unterprogramm INIT alle Variablen mit null initialisiert werden, wird für jeden Verkehrsweg im ersten Durchlauf die Grün-Phasendauer wie bei hoher Verkehrsdichte gewählt.

**Aufgabe A11 (Assembler)**  
(Blatt 3 von 8)

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

```

E068 2A E2E1      PHASE4: LHL    AN3      ;ANKUNFTZEIT WARTESCHLANGE 3
E06B 7C          MOV     A,H      ;TESTE AUF NULL
E06C B5          ORA     L        ;KEINE ANKUNFT = 0
E06D CA CAE0      JZ      PHASE7  ;WEITER
E070 01 1E00      LXI     B,ZW     ;ROT-GELB-DAUER SIGNAL 3
E073 3E 99        MVI     A,99    ;ROT1 + ROT3 + GELB3
E075 D3 00        OUT     AUSGABE
E077 CD 50E1      CALL    DELAY
E07A 01 6400      PHASE5: LXI     B,G3 ;GRUEN-DAUER SIGNAL 3
E07D 2A E4E1      LHL     MARKE3  ;ENDE DER LETZTEN GRUENPHASE
E080 EB          XCHG
E081 2A E2E1      LHL     AN3      ;ANKUNFTZEIT WARTESCHLANGE 1
E084 CD B8E1      CALL    SUB2
E087 C2 93E0      JNZ     AUSG3    ;H-BYTE <> 0
E08A 7D          MOV     A,L      ;TESTE L-BYTE
E08B FE 32        CPI     32      ;KRITERIUM VERKEHRSDICHTE
E08D F2 93E0      JP      AUSG3    ;GERINGE DICHT
E090 CD BFE1      CALL    BMAL2    ;HOHE DICHT
E093 3E A1        AUSG3: MVI     A,0A1 ;ROT1 + GRUEN3
E095 D3 00        OUT     AUSGABE
E097 CD 50E1      CALL    DELAY
E09A 01 0100      PHAS5A: LXI     B,0001 ;1/10 SEKUNDE
E09D CD 50E1      CALL    DELAY
EOA0              ; UEBERPRUEFEN, OB NACHT-SCHALTUNG
EOA0 DB 00        IN      EINGABE ;EINGAENGE LESEN
EOA2 E6 08        ANI     08      ;SCHALTUHR MASKIEREN
EOA4 C2 2FE1      JNZ     PHAS10   ;NUN NACHT-BETRIEB
EOA7              ; UEBERPRUEFEN DER MELDEEINRICHTUNGEN
EOA7 2A DEE1      LHL     AN1      ;NUR WEITER, WENN ANFOR-
EOAA 7C          MOV     A,H      ;DERUNG VORLIEGT.
EOAB B5          ORA     L        ;TESTE AUF NULL
EOAC 2A E6E1      LHL     AN5
EOAF B4          ORA     H
EOB0 B5          ORA     L
EOB1 CA 9AE0      JZ      PHAS5A   ;WARTE
EOB4 01 1E00      PHASE6: LXI     B,ZW ;GELB-DAUER SIGNAL 3
EOB7 3E 91        MVI     A,91    ;ROT1 + GELB3
EOB9 D3 00        OUT     AUSGABE
EOBB CD 50E1      CALL    DELAY
EOBE 21 0000      LXI     H,0000  ;REGISTER H,L = 0
EOC1 22 E2E1      SHLD    AN3      ;ANKUNFTZEIT 3 ZURUECKSETZEN
EOC4 2A EAE1      LHL     ZEIT     ;ENDE GELBPHASE MERKEN
EOC7 22 E4E1      SHLD    MARKE3
EOCA              ;
EOCA              END

```

Fortsetzung mit Teil 2 auf S. 185.

---

## 4. Aufgaben

---

**Aufgabe A11 (Assembler)**  
**(Blatt 4 von 8)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

```
0000      ; AUFGABE 11B, TEIL 2
0000      ; STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
0000      ; ENTSPRECHEND DER VERKEHRSFLUSSDICHTEN
0000      ;
0000      ; KONSTANTEN:
0000      G1      EQU      0096      ;GRUEN-PHASE 1 = 15 S
0000      G3      EQU      0064      ;GRUEN-PHASE 3 = 10 S
0000      G5      EQU      0064      ;GRUEN-PHASE 5 = 10 S
0000      ZW      EQU      001E      ;UEBERGANGSPHASEN = 3 S
0000      ZEITK   EQU      0A25      ;ZEITKONSTANTE 1/10 S
0000      ;
0000      ; DATENBEREICH:
0000      TASTEN  EQU      0E1DD     ;TASTENBYTE
0000      AN1     EQU      0E1DE     ;ANKUNFTZEIT 1
0000      MARKE1  EQU      0E1E0     ;ENDE LETZTE GRUENPHASE 1
0000      AN3     EQU      0E1E2     ;ANKUNFTZEIT 3
0000      MARKE3  EQU      0E1E4     ;ENDE LETZTE GRUENPHASE 3
0000      AN5     EQU      0E1E6     ;ANKUNFTZEIT 5
0000      MARKE5  EQU      0E1E8     ;ENDE LETZTE GRUENPHASE 5
0000      ZEIT    EQU      0E1EA     ;UHRZEIT
0000      ;
0000      ; EIN- UND AUSGABE:
0000      EINGABE EQU      0         ;8-BIT-PARALLEL-EINGABE
0000      AUSGABE EQU      0         ;8-BIT-PARALLEL-AUSGABE
0000      ;
0000      ; ADRESSEN AUS TEIL 1
0000      PHASE1  EQU      0E006
0000      START   EQU      0E00E
0000      ;
0000      ORG      0E0CA      ;STARTADRESSE
E0CA      ;
E0CA 2A E6E1  PHASE7: LHL D AN5      ;ANKUNFTZEIT WARTESCHLANGE 5
E0CD 7C      MOV     A,H          ;TESTE AUF NULL
E0CE B5      ORA     L            ;KEINE ANKUNFT = 0
E0CF CA 06E0  JZ      PHASE1      ;WEITER
E0D2 01 1E00  LXI     B,ZW        ;PAUSEDAUER SIGNAL 5
E0D5 3E 89    MVI     A,89        ;ROT1 + ROT3
E0D7 D3 00    OUT     AUSGABE
E0D9 CD 50E1  CALL    DELAY
E0DC 01 6400  PHASE8: LXI     B,G5  ;GRUEN-DAUER SIGNAL 5
E0DF 2A E8E1  LHL D MARKE5      ;ENDE DER LETZTEN GRUENPHASE
E0E2 EB      XCHG
E0E3 2A E6E1  LHL D AN5          ;ANKUNFTZEIT WARTESCHLANGE 1
E0E6 CD B8E1  CALL    SUB2
E0E9 C2 F5E0  JNZ     AUSG5      ;H-BYTE <> 0
E0EC 7D      MOV     A,L          ;TESTE L-BYTE
E0ED FE 32    CPI     32          ;KRITERIUM VERKEHRSDICHTEN
E0EF F2 F5E0  JP      AUSG5      ;GERINGE DICHTEN
E0F2 CD BFE1  CALL    BMAL2      ;HOHE DICHTEN
E0F5 3E C9    AUSG5: MVI     A,0C9 ;ROT1 + ROT3 + GRUEN5
E0F7 D3 00    OUT     AUSGABE
```

---

#### **4. Aufgaben**

---

**Aufgabe A11 (Assembler)**  
(Blatt 5 von 8)

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

```

E0F9 CD 50E1          CALL    DELAY
E0FC 01 0100          PHAS8A: LXI    B,0001 ;1/10 SEKUNDE
E0FF CD 50E1          CALL    DELAY
E102                  ; UEBERPRUEFEN, OB NACHT-SCHALTUNG
E102 DB 00            IN      EINGABE ;EINGAENGE LESEN
E104 E6 08            ANI     08      ;SCHALTUHR MASKIEREN
E106 C2 2FE1          JNZ     PHAS10 ;NUN NACHT-BETRIEB
E109                  ; UEBERPRUEFEN DER MELDEEINRICHTUNGEN
E109 2A DEE1          LHLD    AN1     ;NUR WEITER, WENN ANFOR-
E10C 7C              MOV     A,H      ;DERUNG VORLIEGT.
E10D B5              ORA     L        ;TESTE AUF NULL
E10E 2A E2E1          LHLD    AN3
E111 B4              ORA     H
E112 B5              ORA     L
E113 CA FCE0          JZ      PHAS8A ;WARTE
E116 01 1E00          PHASE9: LXI    B,ZW ;PAUSEDAUER SIGNAL 5
E119 3E 89            MVI     A,89    ;ROT1 + ROT3
E11B D3 00            OUT     AUSGABE
E11D CD 50E1          CALL    DELAY
E120 21 0000          LXI     H,0000 ;REGISTER H,L = 0
E123 22 E6E1          SHLD    AN5     ;ANKUNFTZEIT 5 ZURUECKSETZEN
E126 2A EAE1          LHLD    ZEIT    ;ENDE GELBPHASE MERKEN
E129 22 E8E1          SHLD    MARKE5
E12C                  ;
E12C C3 06E0          JMP     PHASE1 ;ENDLOSSCHLEIFE
E12F                  ;
E12F                  ; NACHT-BETRIEB
E12F .. .....       PHAS10: ...      ..... ;1/2 SEKUNDE
E132 .. ..           ...      ..... ;GELB1 + GELB3
E134 .. ..           ...      .....
E136 .. .....       ...      .....
E139 .. .....       PHAS11: ...      ..... ;1/2 SEKUNDE
E13C .. ..           ...      ..... ;ALLE LAMPEN AUS
E13E .. ..           ...      .....
E140 .. .....       ...      .....
E143                  ;
E143                  ; UEBERPRUEFEN, OB TAG-SCHALTUNG
E143 .. ..           ..      ..... ;EINGAENGE LESEN
E145 .. ..           ...      ..     ;SCHALTUHR MASKIEREN
E147 .. .....       ...      ..... ;WEITER NACHT-BETRIEB
E14A .. .....       ....      ....  ;DATENBEREICH LOESCHEN
E14D C3 0EE0          JMP     START ;NUN TAG-BETRIEB
E150                  ;
E150                  ; UNTERPROGRAMM DELAY
E150 C5              DELAY: PUSH    B      ;REGISTER RETTEN
E151 D5              PUSH    D
E152 F5              PUSH    PSW
E153 11 250A          DELAY1: LXI    D,ZEITK ;ZEITBASIS 1/10 S
E156 DB 00            DELAY2: IN      EINGABE ;TASTENSTELLUNGEN EINLESEN
E158 F5              PUSH    PSW      ;KOPIE AUF STACK
E159 21 DDE1          LXI     H,TASTEN ;SPEICHER FUER TASTENBYTE

```

---

## 4. Aufgaben

---

**Aufgabe A11 (Assembler)**  
(Blatt 6 von 8)

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

```

E15C AE          XRA      M          ;PRUEFE AUF FLANKENWECHSEL
E15D CA A0E1     JZ       DELAY3     ;NICHTS ZU TUN
E160 F1          POP      PSW
E161 77          MOV      M,A        ;TASTENBYTE SPEICHERN
E162 F5          PUSH     PSW        ;KOPIE AUF STACK
E163 E6 01       TEST1: ANI      01   ;PRUEFE TASTE 1
E165 CA 76E1     JZ       TEST3     ;NICHTS ZU TUN
E168 2A DEE1     LHL      AN1       ;ANKUNFT 1 BEREITS GESETZT?
E16B 7C          MOV      A,H
E16C B5          ORA      L
E16D C2 76E1     JNZ      TEST3     ;BEREITS GESETZT!
E170 2A EAE1     LHL      ZEIT      ;UHRZEIT HOLEN
E173 22 DEE1     SHLD     AN1       ;UND SPEICHERN
E176 F1          TEST3: POP      PSW  ;TASTENBYTE HOLEN
E177 F5          PUSH     PSW        ;UND WIEDER SICHERN
E178 E6 02       ANI      02       ;PRUEFE TASTE 3
E17A CA 8BE1     JZ       TEST5     ;NICHTS ZU TUN
E17D 2A E2E1     LHL      AN3       ;ANKUNFT 3 BEREITS GESETZT?
E180 7C          MOV      A,H
E181 B5          ORA      L
E182 C2 8BE1     JNZ      TEST5     ;BEREITS GESETZT!
E185 2A EAE1     LHL      ZEIT      ;UHRZEIT HOLEN
E188 22 E2E1     SHLD     AN3       ;UND SPEICHERN
E18B F1          TEST5: POP      PSW  ;TASTENBYTE HOLEN
E18C F5          PUSH     PSW        ;UND WIEDER SICHERN
E18D E6 04       ANI      04       ;PRUEFE TASTE 5
E18F CA A0E1     JZ       DELAY3     ;NICHTS ZU TUN
E192 2A E6E1     LHL      AN5       ;ANKUNFT 5 BEREITS GESETZT?
E195 7C          MOV      A,H
E196 B5          ORA      L
E197 C2 A0E1     JNZ      DELAY3     ;BEREITS GESETZT!
E19A 2A EAE1     LHL      ZEIT      ;UHRZEIT HOLEN
E19D 22 E6E1     SHLD     AN5       ;UND SPEICHERN
E1A0 F1          DELAY3: POP      PSW  ;STACK BEREINIGEN
E1A1 1B          DCX      D          ;GEGEN NULL ZAEHLEN
E1A2 7A          MOV      A,D       ;TESTE AUF NULL
E1A3 B3          ORA      E
E1A4 C2 56E1     JNZ      DELAY2     ;SCHLEIFENENDE DELAY2
E1A7 2A EAE1     LHL      ZEIT      ;UHRZEIT HOLEN
E1AA 23          INX      H          ;UM 1 ERHOEHEN
E1AB 22 EAE1     SHLD     ZEIT      ;UND SPEICHERN
E1AE 0B          DCX      B          ;GEGEN NULL ZAEHLEN
E1AF 78          MOV      A,B       ;AUF NULL TESTEN
E1B0 B1          ORA      C
E1B1 C2 53E1     JNZ      DELAY1     ;SCHLEIFENENDE DELAY1
E1B4 F1          POP      PSW        ;REGISTER WIEDER HERSTELLEN
E1B5 D1          POP      D
E1B6 C1          POP      B
E1B7 C9          RET
E1B8             ;

```



---

#### 4. Aufgaben

---

---

**Aufgabe A11 (Assembler)**  
**(Blatt 7 von 8)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

```
E1B8          ; UNTERPROGRAMM SUB2
E1B8          ; (HL) := (HL) - (DE)
E1B8 7D      SUB2:  MOV     A, L
E1B9 93              SUB     E
E1BA 6F              MOV     L, A
E1BB 7C              MOV     A, H
E1BC 9A              SBB     D
E1BD 67              MOV     H, A
E1BE C9              RET
E1BF          ;
E1BF          ; UNTERPROGRAMM BMAL2
E1BF          ; (BC) := (BC) * 2
E1BF 79      BMAL2:  MOV     A, C
E1C0 B7              ORA     A          ; CARRY-BIT LOESCHEN
E1C1 17              RAL
E1C2 4F              MOV     C, A
E1C3 78              MOV     A, B
E1C4 17              RAL
E1C5 47              MOV     B, A
E1C6 C9              RET
E1C7          ;
E1C7          ; UNTERPROGRAMM INIT
E1C7          ; DATENBEREICH LOESCHEN
E1C7 21 0000      INIT:  LXI     H, 0000 ; REGISTER H, L = 0
E1CA 22 DEE1      SHLD     AN1
E1CD 22 E0E1      SHLD     MARKE1
E1D0 22 E2E1      SHLD     AN3
E1D3 22 E4E1      SHLD     MARKE3
E1D6 22 E6E1      SHLD     AN5
E1D9 22 E8E1      SHLD     MARKE5
E1DC C9              RET
E1DD          ;
```

Geben Sie das Programm mit Hilfe des Memory- oder Assembler-Kommandos des MAT 85 komplett ein bzw. ergänzen Sie die Leerstellen-Version des Programms von der Diskette.

Prüfen Sie die korrekte Eingabe mit Hilfe des Print- oder Disassembler-Kommandos.

Starten Sie das Programm mit dem GO-Kommando.

---

#### **4. Aufgaben**

---

---

**Aufgabe A11 (Assembler)**  
(Blatt 8 von 8)

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

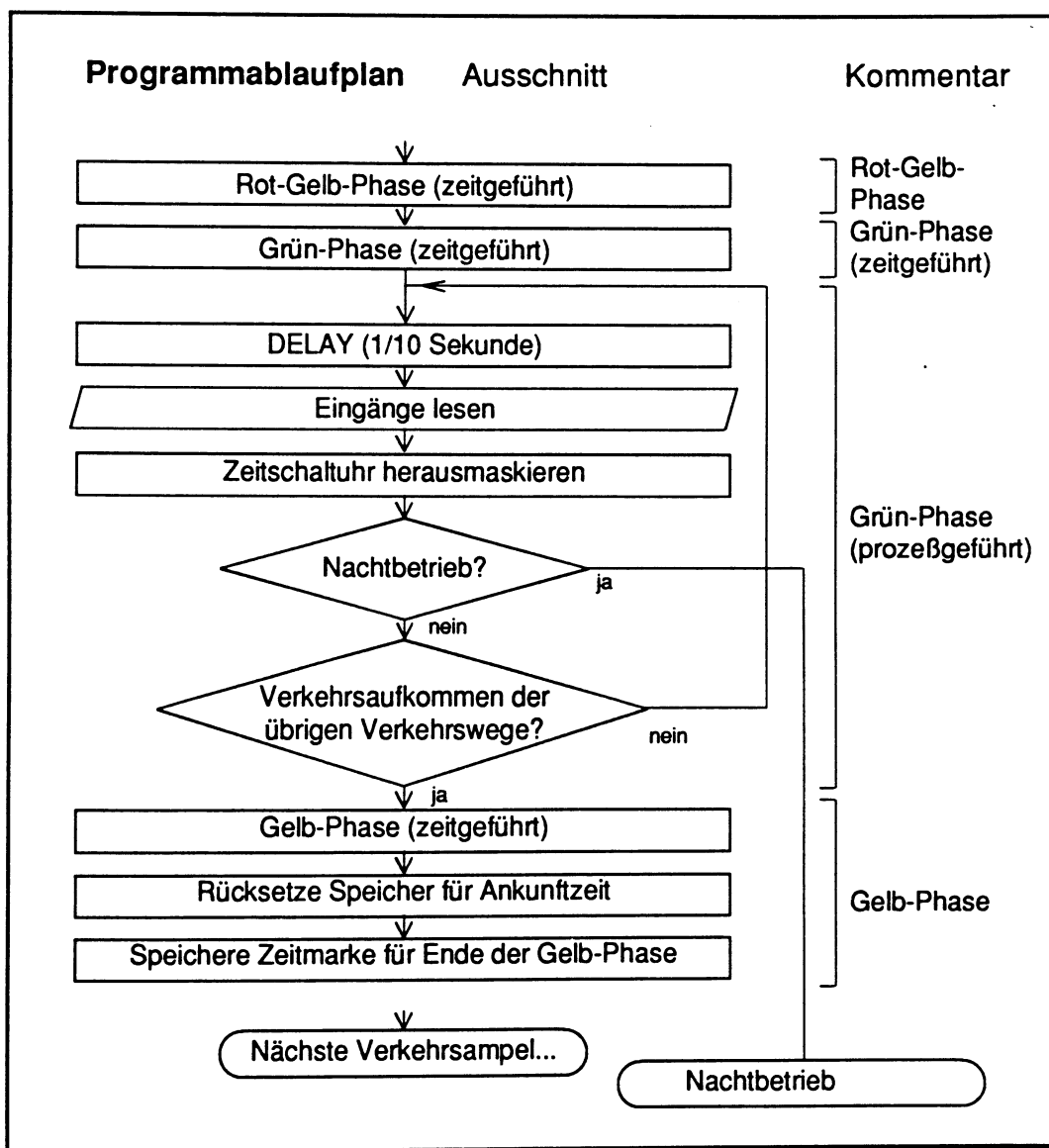
Datum: \_\_\_\_\_

---

**A11c:** Prüfen Sie die Funktionstüchtigkeit des Programms, indem Sie in verschiedenen Situationen zwischen Tag- und Nachtbetrieb hin- und herschalten. Protokollieren Sie Ihre Tests.

## 4.11.2 Betriebsarten-Umschaltung (BASIC)

Die Abfrage der Schaltuhr wird in die Erweiterung der Grün-Phase (prozeßgeführter Schritt) eingefügt. Dies geschieht in allen drei Grün-Phasen, denn sonst könnte es vorkommen, daß bei mangelndem Verkehrsaufkommen die Steuerung in einer Grün-Phase ohne Abfrage verweilt. Der nachstehende Programmablaufplan gilt daher für alle drei Verkehrswege gleichermaßen (das Programm behandelt genauso wie jenes aus A10 alle drei Verkehrswege gleichwertig). Der zweite Abschnitt des Programmablaufplans enthält den Nachtbetrieb zusammen mit dem Übergang zum Tagbetrieb.



---

**Aufgabe A11 (BASIC)**  
**(Blatt 1 von 3)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

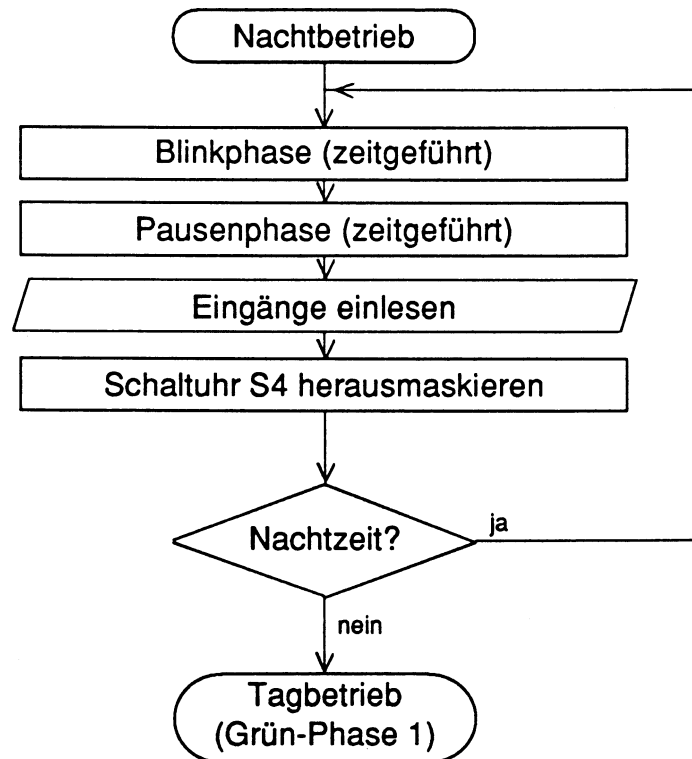
---

**A11b:** Das nachstehende BASIC-Programm führt die Steuerung der Ampeln sowohl im Tag- als auch im Nachtbetrieb durch.

Das Programm ist eine Weiterentwicklung desjenigen aus Aufgabe A10. Es enthält bereits die korrekten Übergangsbedingungen vom Tag- zum Nachtbetrieb. Der Nachtbetrieb und der Übergang zum Tagbetrieb ist noch nicht programmiert. Füllen Sie die Leerstellen im Programm aus.

```
10 REM AUFGABE A11B
20 REM STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
30 REM ENTSPRECHEND DER VERKEHRSFLUSSDICHTEN
31 REM TAG/NACHT-SCHALTUNG GEMAESS SCHALTUHR S4
32 REM STRATEGIE:
33 REM WENN INNERHALB EINES KOMPLETTEN ZYKLUS DIE
34 REM BEDARFSTASTE BZW. INDUKTIONSSCHLEIFE NICHT
35 REM AKTIVIERT WURDE, WERDEN DIE PHASEN FUEER DIESEN
36 REM VERKEHRSWEG AUSGELASSEN.
37 REM WENN INNERHALB VON 5 SEKUNDEN NACH DER GRUEN-
38 REM PHASE DIE BEDARFSTASTE BZW. INDUKTIONSSCHLEIFE
39 REM AKTIVIERT WURDE, LIEGT EINE HOHE VERKEHRSDICHTE
40 REM VOR. DIE DAUER DER GRUENPHASE DIESES VERKEHRSWEGS
41 REM WIRD VERDOPPELT.
42 REM
50 D=150 :REM GRUENPHASE 1 = 15 S
51 F=100 :REM GRUENPHASE 3 = 10 S
52 G=100 :REM GRUENPHASE 5 = 10 S
53 H=30 :REM UEBERGANGSPHASE= 3 S
60 A=0 :REM 8-BIT-PARALLEL-AUSGABE
70 E=0 :REM 8-BIT-PARALLEL-EINGABE
80 Y=0 :REM STARTZEIT = 0
81 K=0 :REM ANKUNFTZEIT 1
82 L=0 :REM ENDE GRUENPHASE 1
83 M=0 :REM ANKUNFTZEIT 3
84 N=0 :REM ENDE GRUENPHASE 3
85 O=0 :REM ANKUNFTZEIT 5
86 P=0 :REM ENDE GRUENPHASE 5
90 GOTO 110 :REM PROGRAMMSTART MIT WEG 1
100 IF K=0 THEN GOTO 400 :REM BEDARF VORHANDEN?
110 Z=H :REM UEBERGANGSPHASE
120 OUT A,139 :REM ROT1 + GELB1 + ROT3
130 GOSUB 1000
200 Z=D :REM GRUENPHASE 1
210 IF K<L+50 THEN Z=2*D :REM VERDOPPELN?
220 OUT A,140 :REM GRUEN1 + ROT3
230 GOSUB 1000
250 Z=1 :REM 1/10 SEKUNDE
260 GOSUB 1000
270 IF (INP(E) AND 8)=8 THEN GOTO 2000
280 IF (M OR O)=0 THEN GOTO 250 :REM BEDARF VORHANDEN?
```

## Programmablaufplan



---

**Aufgabe A11 (BASIC)**  
**(Blatt 2 von 3)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

```
300 Z=H :REM UEBERGANGSPHASE
310 OUT A,138 :REM GELB1 + ROT3
320 GOSUB 1000
330 K=0 :REM ANKUNFTZEIT 1 ZURUECKSETZEN
340 L=Y :REM ENDE GELBPHASE MERKEN
400 IF M=0 THEN GOTO 700 :REM BEDARF VORHANDEN?
410 Z=H :REM UEBERGANGSPHASE
420 OUT A,153 :REM ROT1 + ROT3 + GELB3
430 GOSUB 1000
500 Z=F :REM GRUENPHASE 3
510 IF M<N+50 THEN Z=2*F :REM VERDOPPELN?
520 OUT A,161 :REM ROT1 + GRUEN3
530 GOSUB 1000
550 Z=1 :REM 1/10 SEKUNDE
560 GOSUB 1000
570 IF (INP(E) AND 8)=8 THEN GOTO 2000
580 IF (K OR O)=0 THEN GOTO 550 :REM BEDARF VORHANDEN?
600 Z=H :REM UEBERGANGSPHASE
610 OUT A,145 :REM ROT1 + GELB3
620 GOSUB 1000
630 M=0 :REM ANKUNFTZEIT 3 ZURUECKSETZEN
640 N=Y :REM ENDE GELBPHASE MERKEN
700 IF O=0 THEN GOTO 100 :REM BEDARF VORHANDEN
710 Z=H :REM UEBERGANGSPHASE
720 OUT A,137 :REM ROT1 + ROT3
730 GOSUB 1000
800 Z=G :REM GRUENPHASE 5
810 IF O<P+50 THEN Z=2*G :REM VERDOPPELN?
820 OUT A,201 :REM ROT1 + ROT3 + GRUEN5
830 GOSUB 1000
850 Z=1 :REM 1/10 SEKUNDE
860 GOSUB 1000
870 IF (INP(E) AND 8)=8 THEN GOTO 2000
880 IF (K OR M)=0 THEN GOTO 850 :REM BEDARF VORHANDEN?
900 Z=H :REM UEBERGANGSPHASE
910 OUT A,137 :REM ROT1 + ROT3
920 GOSUB 1000
930 O=0 :REM ANKUNFTZEIT 5 ZURUECKSETZEN
940 P=Y :REM ENDE GELBPHASE MERKEN
999 GOTO 100 :REM GESAMTPROZESS WIEDERHOLEN
1000 REM VERZOEGERUNG (Z = ZEIT IN 1/10 SEKUNDEN)
1010 REM UND AUFZEICHNUNG VON ZEITMARKEN
1020 FOR I=1 TO Z
1030 FOR J=1 TO 2
1040 IF INP(E)=T THEN GOTO 1100
1050 T=INP(E)
1060 IF (T AND 1)=1 THEN IF K=0 THEN K=Y
1070 IF (T AND 2)=2 THEN IF M=0 THEN M=Y
1080 IF (T AND 4)=4 THEN IF O=0 THEN O=Y
1100 NEXT J
1110 Y=Y+1
1120 NEXT I
1130 RETURN
```



**Anmerkung:**

Das Programm weist einen Nachteil auf:

Der Zeittakt von 1/10 Sekunde kann im Unterprogramm VERZOEGERUNG nicht mehr gewahrt bleiben. Die Anzahl der Durchläufe durch die innerste Schleife muß entweder mit 1 (entsprechend einer Taktzeit von ca. 1/15 Sekunden) oder mit 2 (entsprechend einer Taktzeit von ca. 1/7 Sekunde) festgesetzt werden. Im Programmbeispiel wurden zwei Schleifendurchläufe gewählt. Dadurch fallen alle Schaltzeiten entsprechend länger aus; dies ist bei den Zeitmessungen zu berücksichtigen.

Falls die Programmierung unter CP/M mit MBASIC erfolgt, tritt dieser Nachteil nicht auf. Da MBASIC schneller als Steuer-BASIC arbeitet, kann die innere Schleife achtmal durchlaufen werden. Der Abgleich auf 1/10 Sekunde ist gerade noch mit vertretbarer Genauigkeit möglich.

An dieser Stelle zeigt sich, daß die Grenze der Eignung von BASIC für schnelle Schaltvorgänge erreicht wurde.

**Hinweise:**

Da das Programm der Aufgabe A11b einen erheblichen Umfang besitzt, kann zum zügigen Bearbeiten der Aufgabenstellung dem Kursteilnehmer eine Programmvariante (Datei A11BLEER.BAS bzw. A11BLEER.B6K) zur Verfügung gestellt werden, bei der die im Abdruck mit Leerstellen versehenen Zeilen noch nicht ausgefüllt sind.

Zu Programmstart ist die Verkehrsdichte noch nicht definiert. Wenn alle Variablen mit null initialisiert werden, wie dies in den Zeilen 80 bis 86 geschieht, wird im ersten Durchlauf für jeden Verkehrsweg die Grün-Phasendauer wie bei hoher Verkehrsdichte gewählt.

---

**Aufgabe A11 (BASIC)**  
**(Blatt 3 von 3)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

```
2000 REM NACHT-BETRIEB
2010 ... :REM 1/2 SEKUNDE
2020 ... :REM GELB1 + GELB3
2030 .....
2040 ... :REM 1/2 SEKUNDE
2050 ... :REM ALLE LAMPEN AUS
2060 .....
2070 .. .....
2080 GOTO 50 :REM WEITER IM TAG-BETRIEB
```

Geben Sie das Programm komplett ein bzw. ergänzen Sie die Leerstellen-Version des Programms von der Diskette.

Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.

Starten Sie das Programm mit dem Kommando RUN.

**A11c:** Prüfen Sie die Funktionstüchtigkeit des Programms, indem Sie in verschiedenen Situationen zwischen Tag- und Nachtbetrieb hin- und herschalten. Protokollieren Sie Ihre Tests.

**4.11.3 Betriebsarten-Umschaltung (SPS)**

Ein Low-Signal von der Schaltuhr (entsprechend Tageszeit) wird als zusätzliche Übergangsbedingung von der Grün-Phase zur Gelb-Phase eingefügt. Dies gilt für alle drei Grün-Phasen.

Weiterhin wird eine zusätzliche Übergangsbedingung formuliert: Wenn die Schaltuhr ein High-Signal (entsprechend Nachtzeit) meldet und der Software-Timer einer der Grün-Phasen abgelaufen ist, wird der Nachtbetrieb gestartet. Dies geschieht in allen drei Grün-Phasen, denn sonst könnte es vorkommen, daß bei mangelndem Verkehrsaufkommen die Steuerung in einer Grün-Phase ohne Abfrage verweilt.

Die nachstehende Schrittkette gilt für alle drei Verkehrswege gleichermaßen. Das Programm behandelt genauso wie jenes aus A10 alle drei Verkehrswege gleichwertig. Die beiden übrigen Verkehrswege muß man sich sinngemäß ergänzt vorstellen. Die zweite Schrittkette enthält den Nachtbetrieb zusammen mit dem Übergang zum Tagbetrieb.

---

**Aufgabe A11 (SPS)**  
**(Blatt 1 von 7)**

---

Name: 

---

Kurs: 

---

Datum: 

---

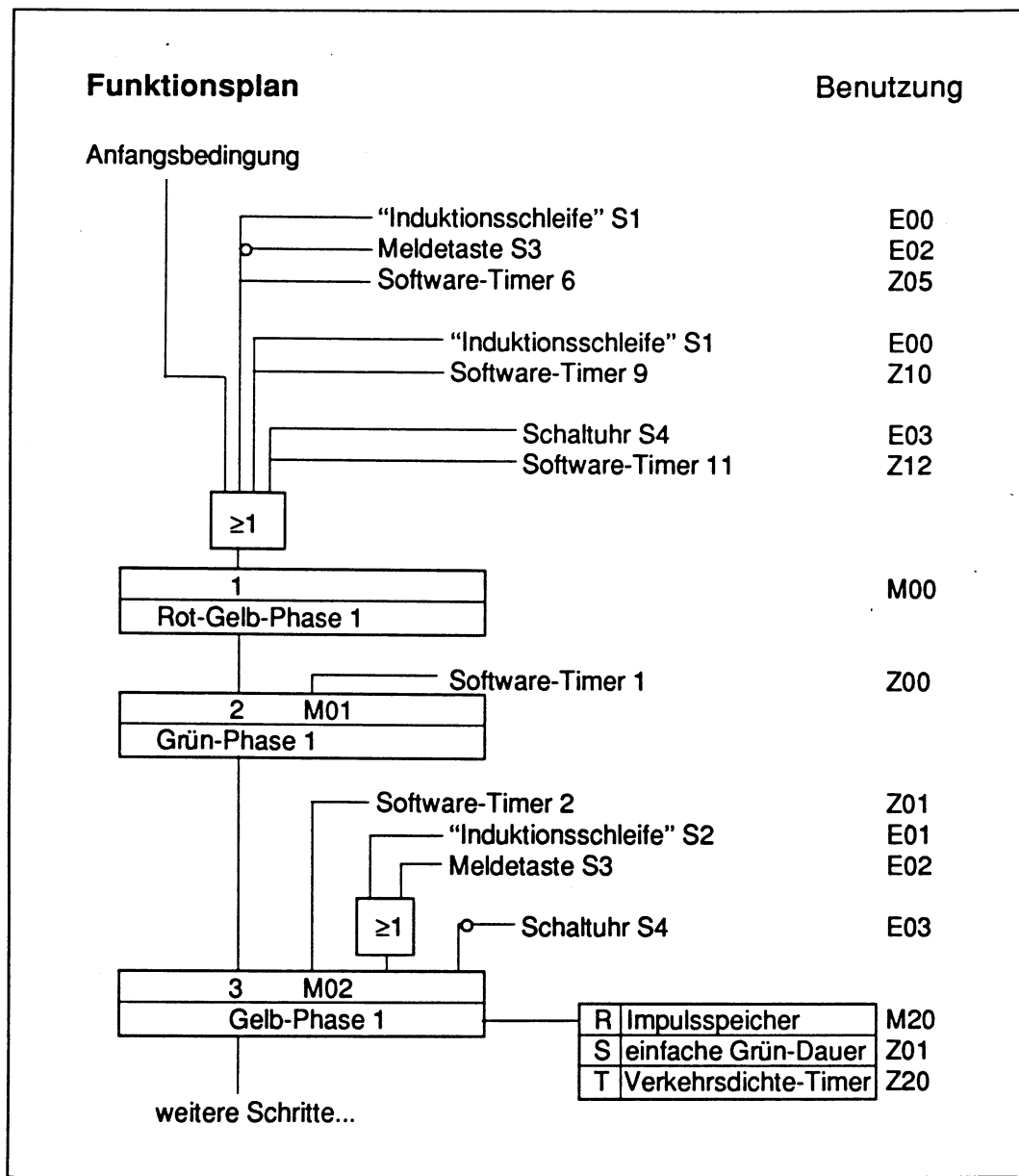
**A11b:** Das nachstehende SPS-Programm führt die Steuerung der Ampeln sowohl im Tag- als auch im Nachtbetrieb durch.

Das Programm ist eine Weiterentwicklung desjenigen aus Aufgabe A10. Es enthält bereits eine korrekte Übergangsbedingung vom Tag- zum Nachtbetrieb. Der Nachtbetrieb und der Übergang zum Tagbetrieb ist noch nicht programmiert. Füllen Sie die Leerstellen im Programm aus.

```
; AUFGABE A11B
; STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
; ENTSPRECHEND DER VERKEHRSFLUSSDICHTEN
; TAG/NACHT-SCHALTUNG GEMAESS SCHALTUHR S4

; STRATEGIE:
; WENN INNERHALB EINES KOMPLETTEN ZYKLUS DIE
; BEDARFSTASTE BZW. INDUKTIONSSCHLEIFE NICHT
; AKTIVIERT WURDE, WERDEN DIE PHASEN FÜR DIESEN
; VERKEHRSWEG AUSGELASSEN.
; WENN INNERHALB VON 5 SEKUNDEN NACH DER GELB- BZW.
; PAUSENPHASE DIE BEDARFSTASTE BZW. INDUKTIONSSCHLEIFE
; AKTIVIERT WURDE, LIEGT EINE HOHE VERKEHRSDICHTE VOR.
; DIE DAUER DER GRÜNPHASE DIESES VERKEHRSWEGS WIRD
; IM NÄCHSTEN ZYKLUS VERDOPPELT.
; INITIALISIEREN DER ZEITGEBER

UN M 00      ; ANFANGSBEDINGUNG:
UN M 01      ; ALLE MERKER NICHT
UN M 02      ; GESETZT.
UN M 03      ;
UN M 04      ;
UN M 05      ;
UN M 06      ;
UN M 07      ;
UN M 10      ;
UN M 11      ;
UN M 12      ;
=L Z 00,30    ;1. ROT-GELB-PHASE AMPEL 1
=L Z 01,150   ;2. GRÜN-PHASE AMPEL 1
=L Z 02,30    ;3. GELB-PHASE AMPEL 1
=L Z 03,30    ;4. ROT-GELB-PHASE AMPEL 3
=L Z 04,100   ;5. GRÜN-PHASE AMPEL 3
=L Z 05,30    ;6. GELB-PHASE AMPEL 3
=L Z 06,30    ;7. PAUSENPHASE AMPEL 5
=L Z 07,100   ;8. GRÜNPHASE AMPEL 5
=L Z 10,30    ;9. PAUSENPHASE AMPEL 5
=L Z 11,5     ; BLINKDAUER GELB
=L Z 12,5     ; PAUSENDAUER GELB
=L Z 20,50    ; ZEITINTERVALL VERKEHRSDICHTE
=S A 07      ; FUSSGAENGERAMPELN FREIGEBEN
=S M 00      ;
=S Z 00      ;1 STARTE PHASE 1
```



---

**Aufgabe A11 (SPS)**  
**(Blatt 2 von 7)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

; ABLAUFKETTE (GESTEUERT DURCH ZEITGEBER)

```
U   Z 00      ;   PHASE 1 ABGELAUFEN
=R  Z 00      ;
=R  M 00      ;
=S  Z 01      ;   STARTE PHASE 2
=S  M 01      ;2

U   M 21      ;   WENN FAHRZEUG AN IND.SCHL. 3
O   M 22      ;   ODER FUSSGAENGER AN TASTE 5
=   M 23      ;3  HILFSMERKER

U   M 23      ;   S.O.
UN  E 03      ;   UND TAGESZEIT
U   Z 01      ;   UND PHASE 2 ABGELAUFEN
=R  Z 01      ;
=R  M 01      ;
=S  Z 02      ;   STARTE PHASE 3
=S  M 02      ;4

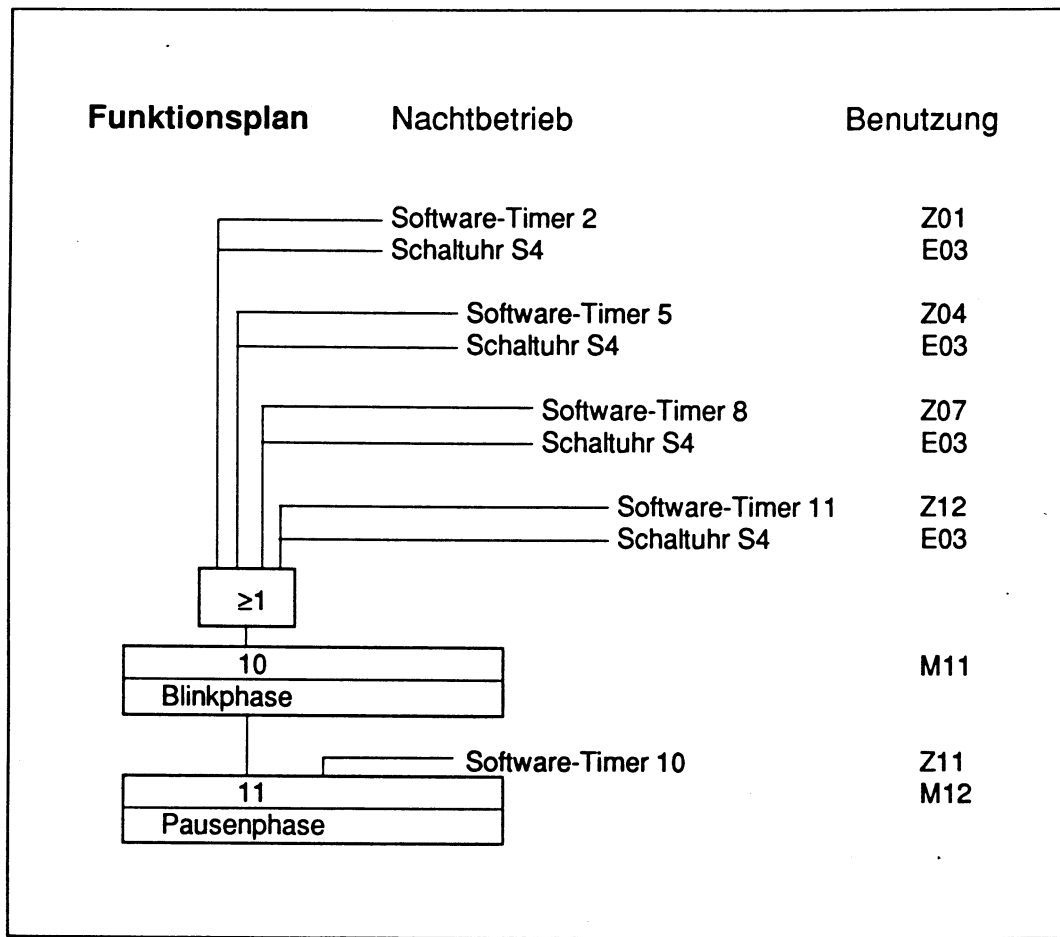
U   Z 02      ;   WENN PHASE 3 ABGELAUFEN
=R  M 20      ;   RUECKSETZE IMPULSSPEICHER DER IND.SCHL. 1
=L  Z 01,150  ;   GRUENPHASE AMPEL 1 = 15 SEKUNDEN
=S  Z 20      ;   STARTE VERKEHRSDICHTE-TIMER
=S  M 30      ;5  MERKER MESSPHASE AMPEL 1

U   M 21      ;   WENN FAHRZEUG AN IND.SCHL. 3
U   Z 02      ;   UND PHASE 3 ABGELAUFEN
=R  Z 02      ;
=R  M 02      ;
=S  Z 03      ;   STARTE PHASE 4
=S  M 03      ;6

U   M 22      ;   WENN FUSSGAENGER AN TASTE 5
UN  M 21      ;   UND KEIN FAHRZEUG AN IND.SCHL. 3
U   Z 02      ;   UND PHASE 3 ABGELAUFEN
=R  Z 02      ;
=R  M 02      ;
=S  Z 06      ;   STARTE PHASE 7
=S  M 06      ;7

U   Z 03      ;   PHASE 4 ABGELAUFEN
=R  Z 03      ;
=R  M 03      ;
=S  Z 04      ;   STARTE PHASE 5
=S  M 04      ;8

U   M 20      ;   WENN FAHRZEUG AN IND.SCHL. 1
O   M 22      ;   ODER FUSSGAENGER AN TASTE 5
=   M 24      ;9  HILFSMERKER
```



**Aufgabe A11 (SPS)**  
**(Blatt 3 von 7)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

```
U   M 24      ;   S.O.
UN  E 03      ;   UND TAGESZEIT
U   Z 04      ;   UND PHASE 5 ABGELAUFEN
=R  Z 04      ;
=R  M 04      ;
=S  Z 05      ;   STARTE PHASE 6
=S  M 05      ;10

U   Z 05      ;   WENN PHASE 6 ABGELAUFEN
=R  M 21      ;   RUECKSETZE IMPULSSPEICHER DER IND.SCHL. 3
=L  Z 04,100  ;   GRUENPHASE AMPEL 3 = 10 SEKUNDEN
=S  Z 20      ;   STARTE VERKEHRSDICHTE-TIMER
=S  M 31      ;11 MERKER MESSPHASE AMPEL 3

U   M 22      ;   WENN FUSSGAENGER AN TASTE 5
U   Z 05      ;   UND PHASE 6 ABGELAUFEN
=R  Z 05      ;
=R  M 05      ;
=S  Z 06      ;   UND STARTE PHASE 7
=S  M 06      ;12

U   M 20      ;   WENN FAHRZEUG AN IND.SCHL. 1
UN  M 22      ;   UND KEIN FUSSGAENGER AN TASTE 5
U   Z 05      ;   UND PHASE 6 ABGELAUFEN
=R  Z 05      ;
=R  M 05      ;
=S  Z 00      ;   STARTE PHASE 1
=S  M 00      ;13

U   Z 06      ;   PHASE 7 ABGELAUFEN
=R  Z 06      ;
=R  M 06      ;
=S  Z 07      ;   STARTE PHASE 8
=S  M 07      ;14

U   M 20      ;   WENN FAHRZEUG AN IND.SCHL. 1
O   M 21      ;   ODER FAHRZEUG AN IND.SCHL. 3
=   M 25      ;15 HILFSMERKER

U   M 25      ;   S.O.
UN  E 03      ;   UND TAGESZEIT
U   Z 07      ;   UND PHASE 8 ABGELAUFEN
=R  Z 07      ;
=R  M 07      ;
=S  Z 10      ;   STARTE PHASE 9
=S  M 10      ;16

U   Z 10      ;   WENN PHASE 9 ABGELAUFEN
=R  M 22      ;   RUECKSETZE IMPULSSPEICHER TASTE 5
=L  Z 07,100  ;   GRUENPHASE AMPEL 5 = 10 SEKUNDEN
=S  Z 20      ;   STARTE VERKEHRSDICHTE-TIMER
=S  M 32      ;17 MERKER MESSPHASE AMPEL 5
```



**Hinweis:**

Wenn der Kursteilnehmer den BFZ-SPS-Interpreter in der Version der FTÜ benutzt, so wird von nachstehendem gleichwertigen Programm ausgegangen.

```
*/M00*/M01*/M02*/M03*/M04*/M05*/M06*/M07*/M10*/M11*/M12=LZ00,30
=LZ01,150=LZ02,30=LZ03,30=LZ04,100=LZ05,30=LZ06,30=LZ07,100
=LZ10,30=LZ11,5=LZ12,5=LZ20,50=SA07=SM00=SZ00
*Z00=RZ00=RM00=SZ01=SM01
*M21+M22=M23
*M23*/E03*Z01=RZ01=RM01=SZ02=SM02
*Z02=RM20=LZ01,150=SZ20=SM30
*M21*Z02=RZ02=RM02=SZ03=SM03
*M22*/M21*Z02=RZ02=RM02=SZ06=SM06
*Z03=RZ03=RM03=SZ04=SM04
*M20+M22=M24
*M24*/E03*Z04=RZ04=RM04=SZ05=SM05
*Z05=RM21=LZ04,100=SZ20=SM31
*M22*Z05=RZ05=RM05=SZ06=SM06
*M20*/M22*Z05=RZ05=RM05=SZ00=SM00
*Z06=RZ06=RM06=SZ07=SM07
*M20+M21=M25
*M25*/E03*Z07=RZ07=RM07=SZ10=SM10
*Z10=RM22=LZ07,100=SZ20=SM32
*M20*Z10=RZ10=RM10=SZ00=SM00
*M21*/M20*Z10=RZ10=RM10=SZ03=SM03
*E00=SM20
*E01=SM21
*E02=SM22
*Z20=RZ20=RM30=RM31=RM32
*M20*M30=LZ01,300
*M21*M31=LZ04,200
*M22*M32=LZ07,200
*E03*Z01=RZ01=RM01=RA07=SZ11=SM11
.....
.....
.....
.....
.....
*M00+M03+M04+M05+M06+M07+M10=A00
*M00+M02+M11=A01
*M01=A02
*M00+M01+M02+M03+M06+M07+M10=A03
*M03+M05+M11=A04
*M04=A05
*M07=A06
```

---

**Aufgabe A11 (SPS)**  
**(Blatt 4 von 7)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

```
U   M 20      ;   WENN FAHRZEUG AN IND.SCHL. 1
U   Z 10      ;   UND PHASE 9 ABGELAUFEN
=R  Z 10      ;
=R  M 10      ;
=S  Z 00      ;   STARTE PHASE 1
=S  M 00      ;18
```

```
U   M 21      ;   WENN FAHRZEUG AN IND.SCHL. 3
UN  M 20      ;   UND KEIN FAHRZEUG AN IND.SCHL. 1
U   Z 10      ;   UND PHASE 9 ABGELAUFEN
=R  Z 10      ;
=R  M 10      ;
=S  Z 03      ;   STARTE PHASE 4
=S  M 03      ;19
```

; SPEICHERUNG DER EINGABEIMPULSE

```
U   E 00      ;   INDUKTIONSSCHLEIFE 1
=S  M 20      ;20
```

```
U   E 01      ;   INDUKTIONSSCHLEIFE 3
=S  M 21      ;21
```

```
U   E 02      ;   FUSSGAENGERTASTE 5
=S  M 22      ;22
```

; VERKEHRSDICHTENSTEUERUNG

```
U   Z 20      ;   WENN VERKEHRSDICHTE-TIMER ABGELAUFEN
=R  Z 20      ;   WIEDER ZURUECKSETZEN
=R  M 30      ;   HILFSMERKER EBENFALLS
=R  M 31      ;   ZURUECKSETZEN
=R  M 32      ;23
```

```
U   M 20      ;   WENN IMPULSSPEICHER IND.SCHL. 1
U   M 30      ;   UND VERKEHRSDICHTE-TIMER
=L  Z 01,300   ;24 GRUENPHASE AMPEL 1 = 30 SEKUNDEN
```

```
U   M 21      ;   WENN IMPULSSPEICHER IND.SCHL. 3
U   M 31      ;   UND VERKEHRSDICHTE-TIMER
=L  Z 04,200   ;25 GRUENPHASE AMPEL 3 = 20 SEKUNDEN
```

```
U   M 22      ;   WENN IMPULSSPEICHER TASTE 5
U   M 32      ;   UND VERKEHRSDICHTE-TIMER
=L  Z 07,200   ;26 GRUENPHASE AMPEL 5 = 20 SEKUNDEN
```

**Hinweis:**

Da das Programm der Aufgabe A11b einen erheblichen Umfang besitzt, kann zum zügigen Bearbeiten der Aufgabenstellung dem Kursteilnehmer eine Programmvariante (Datei A11BLEER.DOK bzw. A11BLEER.DIN bzw. A11BLEER.SPS) zur Verfügung gestellt werden, bei der die im Abdruck mit Leerstellen versehenen Zeilen noch nicht ausgefüllt sind.

---

**Aufgabe A11 (SPS)**  
**(Blatt 5 von 7)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

; TAG/NACHT-SCHALTUNG

```
U   E 03      ;   WENN NACHTZEIT
U   Z 01      ;   UND PHASE 2 ABGELAUFEN
=R  Z 01      ;
=R  M 01      ;
=R  A 07      ;   FUSSGAENGERAMPEN ABSCHALTEN
=S  Z 11      ;   STARTE BLINKPHASE GELB
=S  M 11      ;27

.   .  . .    ;   WENN NACHTZEIT
.   .  . .    ;   UND PHASE 5 ABGELAUFEN
..  .  . .    ;
..  .  . .    ;
..  .  . .    ;   FUSSGAENGERAMPEN ABSCHALTEN
..  .  . .    ;   STARTE BLINKPHASE GELB
..  .  . .    ;28

.   .  . .    ;   WENN NACHTZEIT
.   .  . .    ;   UND PHASE 8 ABGELAUFEN
..  .  . .    ;
..  .  . .    ;
..  .  . .    ;   FUSSGAENGERAMPEN ABSCHALTEN
..  .  . .    ;   STARTE BLINKPHASE GELB
..  .  . .    ;29

.   .  . .    ;   WENN BLINKPHASE GELB ABGELAUFEN
..  .  . .    ;
..  .  . .    ;
..  .  . .    ;   STARTE PAUSENPHASE GELB
..  .  . .    ;30

.   .  . .    ;   WENN NACHTZEIT
.   .  . .    ;   UND PAUSENPHASE GELB ABGELAUFEN
..  .  . .    ;
..  .  . .    ;
..  .  . .    ;   STARTE BLINKPHASE GELB
..  .  . .    ;31

..  .  . .    ;   WENN TAGESZEIT
.   .  . .    ;   UND PAUSENPHASE GELB ABGELAUFEN
..  .  . .    ;
..  .  . .    ;
..  .  . .    ;   FUSSGAENGERAMPEN EINSCHALTEN
..  .  . .    ;   SPEICHER DER MELDEIMPULSE LOESCHEN
..  .  . .    ;
..  .  . .    ;
..  .  . .    ;   STARTE PHASE 1
..  .  . .    ;32
```

---

## 4. Aufgaben

---

---

**Aufgabe A11 (SPS)**  
**(Blatt 6 von 7)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

Datum: \_\_\_\_\_

---

; AUSGABESTEuerung

```
U  M 00      ;   IN  PHASE 1
O  M 03      ;   UND PHASE 4
O  M 04      ;   UND PHASE 5
O  M 05      ;   UND PHASE 6
O  M 06      ;   UND PHASE 7
O  M 07      ;   UND PHASE 8
O  M 10      ;   UND PHASE 9
=  A 00      ;33 ROT1 SCHALTEN

U  M 00      ;   IN  PHASE 1
O  M 02      ;   UND PHASE 3
O  M 11      ;   UND BLINKPHASE GELB
=  A 01      ;34 GELB1 SCHALTEN

U  M 01      ;   IN  PHASE 2
=  A 02      ;35 GRUEN1 SCHALTEN

U  M 00      ;   IN  PHASE 1
O  M 01      ;   UND PHASE 2
O  M 02      ;   UND PHASE 3
O  M 03      ;   UND PHASE 4
O  M 06      ;   UND PHASE 7
O  M 07      ;   UND PHASE 8
O  M 10      ;   UND PHASE 9
=  A 03      ;36 ROT3 SCHALTEN

U  M 03      ;   IN  PHASE 4
O  M 05      ;   UND PHASE 6
O  M 11      ;   UND BLINKPHASE GELB
=  A 04      ;37 GELB3 SCHALTEN

U  M 04      ;   IN PHASE 5
=  A 05      ;38 GRUEN3 SCHALTEN

U  M 07      ;   IN PHASE 8
=  A 06      ;39 GRUEN5 SCHALTEN

END
```

Geben Sie das Programm komplett ein bzw. ergänzen Sie die Leerstellen-Version des Programms von der Diskette.

Prüfen Sie die korrekte Eingabe mit Hilfe des LIST-Kommandos.

Starten Sie das Programm mit dem Kommando GO.

---

## 4. Aufgaben

---

---

**Aufgabe A11 (SPS)**  
**(Blatt 7 von 7)**

Name: \_\_\_\_\_

Kurs: \_\_\_\_\_

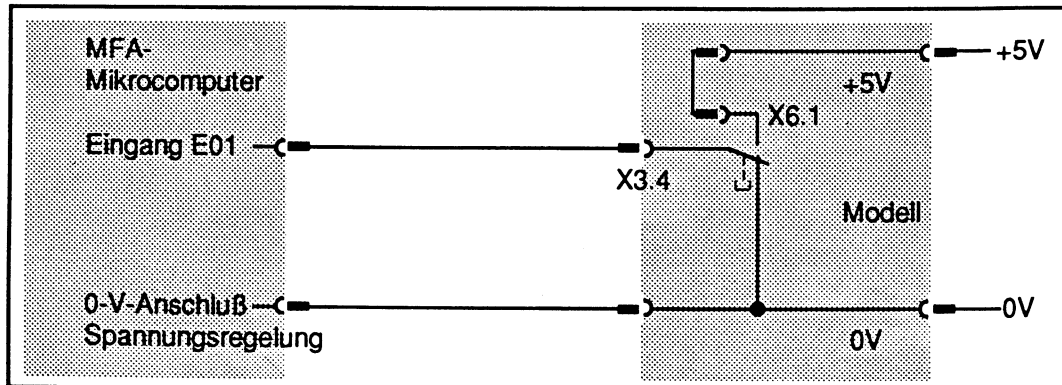
Datum: \_\_\_\_\_

---

**A11c:** Prüfen Sie die Funktionstüchtigkeit des Programms, indem Sie in verschiedenen Situationen zwischen Tag- und Nachtbetrieb hin- und herschalten. Protokollieren Sie Ihre Tests.



**A1a:** Zeichnen Sie den Stromlaufplan für S2.



Beantworten Sie folgende Fragen:

**A1b:** Welche Funktion hat S1?

**S1 meldet die Ankunft eines Fahrzeugs auf der Hauptstraße bei der Ampel 1.**

**A1c:** Welche Funktion hat S3?

**S3 meldet, daß ein Fußgänger am Fußgängerübergang 5 die Straße überqueren möchte.**

**A1d:** Welche Funktion haben H1, H2 und H3?

**H1, H2 und H3 sind die rote, gelbe und grüne Lampe der Verkehrsampel 1. Sie regeln den Fahrzeugverkehr auf der Hauptstraße.**

**A1e:** Welche Funktion haben H4 und H5?

**H4 und H5 sind die rote und die grüne Lampe der Fußgängerampel 2. Sie regeln das Überqueren der Nebenstraße für die Fußgänger.**

**A2a:** Um nur H3 (grüne Lampe der Verkehrsampel 1) einzuschalten, muß mit Hilfe des OUT-Kommandos ausgegeben werden:

**04**

**A2b:** Um nur H7 (gelbe Lampe der Verkehrsampel 3) einzuschalten, muß mit Hilfe des OUT-Kommandos ausgegeben werden:

**10** (hex)

**A2c:** Um H1 und H2 (rote und gelbe Lampe der Verkehrsampel 1) einzuschalten, muß mit Hilfe des OUT-Kommandos ausgegeben werden:

**03**

**A3a:** Welche Funktion wird durch den Ausgang A07 gesteuert?

**Mit Ausgang A07 wird die Betriebsbereitschaft der Fußgängerampeln gesteuert.**

Um nur A07 auf High zu schalten, muß eingegeben werden:

DATEN = 00 80      80 SP eingeben.

Beschreiben Sie Ihre Beobachtung bez. der Fußgängerampeln 2, 4 und 5.

**Wenn A07 High-Potential führt, sind die Fußgängerampeln eingeschaltet und können auf Rot oder Grün geschaltet werden.**

**Wenn A07 Low-Potential führt, sind die Fußgängerampeln ausgeschaltet (dunkel).**

**A3b:** Um die Fußgängerampel 5 auf rot zu schalten, muß welches Datenwort ausgegeben werden?

**C0** (hex)

**A3c:** Wie hängen die Verkehrsampel 1 und die Fußgängerampel 2 zusammen? Um die Fußgängerampel 2 auf grün zu schalten, muß welches Datenwort ausgegeben werden?

**Die Fußgängerampel 2 ist immer dann auf Grün geschaltet, wenn die Verkehrsampel 1 ebenfalls auf Grün geschaltet ist. Fußgängerampel 2 wird durch Ausgabe von 84 (hex) auf Grün geschaltet.**

**A3d:** Wie hängen die Verkehrsampel 3 und die Fußgängerampel 4 zusammen? Um die Fußgängerampel 4 auf grün zu schalten, muß welches Datenwort ausgegeben werden?

**Die Fußgängerampel 4 ist immer dann auf Grün geschaltet, wenn die Verkehrsampel 3 ebenfalls auf Grün geschaltet ist. Fußgängerampel 4 wird durch Ausgabe von A0 (hex) auf Grün geschaltet.**

**A4a:** Bringen Sie die Schalter der Eingänge E00 bis E02 in die linke Stellung und die Schalter der Eingänge E03 bis E07 in die rechte Stellung. Alle Leuchtdioden sind dunkel. Warum?

**E00 bis E02 werden durch S1 bis S3 gesteuert; diese liefern in Ruhestellung Low-Potential. E03 bis E07 werden durch die Stellung der Schalter der 8-Bit-Parallel-Eingabe ebenfalls auf Low-Potential gelegt.**

**A4b:** Drücken Sie nur den Taster S1. Welche Leuchtdiode leuchtet? Notieren Sie die Bildschirmanzeige und unterstreichen Sie diejenige Dualstelle, die sich bei Tastendruck ändert.

**Die Leuchtdiode von E00 leuchtet. Bildschirmanzeige: 00000001**

**A4c:** Drücken Sie nur den Taster S2. Welche Leuchtdiode leuchtet? Notieren Sie die Bildschirmanzeige und unterstreichen Sie diejenige Dualstelle, die sich bei Tastendruck ändert.

**Die Leuchtdiode von E01 leuchtet. Bildschirmanzeige: 00000010**

**A4d:** Drücken Sie nur den Taster S3. Welche Leuchtdiode leuchtet? Notieren Sie die Bildschirmanzeige und unterstreichen Sie diejenige Dualstelle, die sich bei Tastendruck ändert.

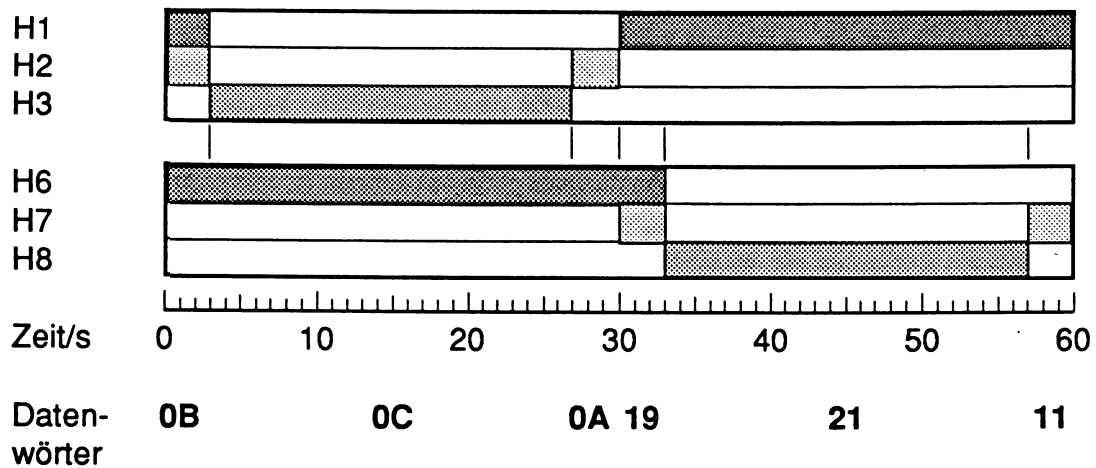
**Die Leuchtdiode von E02 leuchtet. Bildschirmanzeige: 00000100**

**A4e:** Fassen Sie zusammen:

**In Ruhestellung erzeugen die Melde-Einrichtungen Low-Signal.**

**Wenn betätigt, erzeugen die Melde-Einrichtungen High-Signal.**

**A5a:** Füllen Sie folgendes Signal-Zeit-Diagramm aus. Schraffieren Sie (vorzugsweise mit Farbstiften) die Zeitabschnitte, in denen die Lampen leuchten. Beginnen Sie mit der Rot-Gelb-Phase der Verkehrsampel 1. Schreiben Sie die Datenwörter, die zu den jeweiligen Abschnitten gehören, unter das Signal-Zeit-Diagramm. Beachten Sie, daß die Fußgängerampeln dunkel bleiben sollen.



**A5b:** Geben Sie die Datenwörter der Reihenfolge nach mit Hilfe des Out-Kommandos des MAT 85 aus. Kontrollieren Sie die Richtigkeit der Ampelanzeigen.

**A5c:** Füllen Sie die Leerstellen in dem folgenden Assemblerprogramm durch Konstanten aus. Die Konstanten nach dem LXI-Befehl sind die festgelegten Zeitdauern (Zwei-Byte-Konstanten) und die Konstanten nach dem MVI-Befehl die zuvor ermittelten Datenwörter. Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln entsprechend dem Signal-Zeit-Diagramm in Aufgabe A5a durchführen.

```
0000          ; AUFGABE A5C
0000          ; STEUERUNG DES FAHRZEUGVERKEHRS
0000          ;
0000          EINGABE EQU    0          ;8-BIT-PARALLEL-EINGABE
0000          AUSGABE EQU    0          ;8-BIT-PARALLEL-AUSGABE
0000          ;
0000          ORG      0E000    ;STARTADRESSE
E000          ;
E000 01 1E00   PHASE1: LXI      B,001E    ;3 SEKUNDEN
E003 3E 0B     MVI      A,0B      ;ROT1 + GELB1 + ROT3
E005 D3 00     OUT      AUSGABE
E007 CD 3FE0   CALL     DELAY
E00A 01 F000   PHASE2: LXI      B,00F0    ;24 SEKUNDEN
E00D 3E 0C     MVI      A,0C      ;GRUEN1 + ROT3
E00F D3 00     OUT      AUSGABE
E011 CD 3FE0   CALL     DELAY
E014 01 1E00   PHASE3: LXI      B,001E    ;3 SEKUNDEN
E017 3E 0A     MVI      A,0A      ;GELB1 + ROT3
E019 D3 00     OUT      AUSGABE
E01B CD 3FE0   CALL     DELAY
E01E 01 1E00   PHASE4: LXI      B,001E    ;3 SEKUNDEN
E021 3E 19     MVI      A,19      ;ROT1 + ROT3 + GELB3
E023 D3 00     OUT      AUSGABE
E025 CD 3FE0   CALL     DELAY
E028 01 F000   PHASE5: LXI      B,00F0    ;24 SEKUNDEN
E02B 3E 21     MVI      A,21      ;ROT1 + GRUEN3
E02D D3 00     OUT      AUSGABE
E02F CD 3FE0   CALL     DELAY
E032 01 1E00   PHASE6: LXI      B,001E    ;3 SEKUNDEN
E035 3E 11     MVI      A,11      ;ROT1 + GELB3
E037 D3 00     OUT      AUSGABE
E039 CD 3FE0   CALL     DELAY
E03C          ;
E03C C3 00E0   JMP      PHASE1    ;ENDLOSSCHLEIFE
E03F          ;
```

Restliche Teile des Programms wie in der Aufgabenstellung.

- ▲ **A5d:** Füllen Sie die Leerstellen im Datenbereich des folgenden Assemblerprogramms durch Konstanten aus. Die Konstanten nach den DW-Assembleranweisungen sind die festgelegten Zeitdauern (Zwei-Byte-Konstanten) und die Konstanten nach den DB-Assembleranweisungen die zuvor ermittelten Datenwörter. Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln entsprechend dem Signal-Zeit-Diagramm in Aufgabe A5a durchführen.

```
E02D      ;  
E02D      ; TABELLE DER AMPELPHASEN  
E02D 0B      TAB:  DB      0B      ;ROT1 + GELB1 + ROT3  
E02E 1E00      DW      001E      ;3 SEKUNDEN  
E030 0C      DB      0C      ;GRUEN1 + ROT3  
E031 F000      DW      00F0      ;24 SEKUNDEN  
E033 0A      DB      0A      ;GELB1 + ROT3  
E034 1E00      DW      001E      ;3 SEKUNDEN  
E036 19      DB      19      ;ROT1 + ROT3 + GELB3  
E037 1E00      DW      001E      ;3 SEKUNDEN  
E039 21      DB      21      ;ROT1 + GRUEN3  
E03A F000      DW      00F0      ;24 SEKUNDEN  
E03C 11      DB      11      ;ROT1 + GELB3  
E03D 1E00      DW      001E      ;3 SEKUNDEN  
E03F      ;
```

Restliche Teile des Programms wie in der Aufgabenstellung.

**A5c:** Füllen Sie die Leerstellen in dem folgenden BASIC-Programm durch Konstanten aus. Die Konstanten in der Zuweisung Z=... sind die festgelegten Zeitdauern und die Konstanten nach dem OUT-Befehl die zuvor ermittelten Datenwörter. Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln entsprechend dem Signal-Zeit-Diagramm in Aufgabe A5a durchführen.

```
10 REM AUFGABE A5C
20 REM STEUERUNG DES FAHRZEUGVERKEHRS
30 REM
40 A=0                                :REM 8-BIT-PARALLEL-AUSGABE
100 Z=30                              :REM 3 SEKUNDEN
110 OUT A,11                          :REM ROT1 + GELB1 + ROT3
120 GOSUB 1000
200 Z=240                             :REM 24 SEKUNDEN
210 OUT A,12                          :REM GRUEN1 + ROT3
220 GOSUB 1000
300 Z=30                              :REM 3 SEKUNDEN
310 OUT A,10                          :REM GELB1 + ROT3
320 GOSUB 1000
400 Z=30                              :REM 3 SEKUNDEN
410 OUT A,25                          :REM ROT1 + ROT3 + GELB3
420 GOSUB 1000
500 Z=240                             :REM 24 SEKUNDEN
510 OUT A,33                          :REM ROT1 + GRUEN3
520 GOSUB 1000
600 Z=30                              :REM 3 SEKUNDEN
610 OUT A,17                          :REM ROT1 + GELB3
620 GOSUB 1000
999 GOTO 100
1000 REM VERZOEGERUNG (Z = ZEIT IN 1/10 SEKUNDE)
1010 FOR I=1 TO Z
1020 FOR J=1 TO 98
1030 NEXT J
1040 NEXT I
1050 RETURN
```



- ▲ **A5d:** Füllen Sie die Leerstellen in den DATA-Zeilen des folgenden BASIC-Programms durch Konstanten aus. Eine DATA-Zeile steht immer für eine Ampelphase (dies ist keine Notwendigkeit, sondern dient der Übersichtlichkeit). Die erste Konstante in einer Zeile ist jeweils eines der zuvor ermittelten Datenwörter. Die zweite Konstante in einer Zeile ist die dazugehörige Phasendauer in 1/10-Sekunden. Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln entsprechend dem Signal-Zeit-Diagramm in Aufgabe A5a durchführen.

```
2000 REM DATENBEREICH
2010 DATA 11,30           :REM ROT1 + GELB1 + ROT3, 3 SEK.
2020 DATA 12,240         :REM GRUEN1 + ROT3, 24 SEK.
2030 DATA 10,30          :REM GELB1 + ROT3
2040 DATA 25,30          :REM ROT1 + ROT3 + GELB3, 3 SEK.
2050 DATA 33,240         :REM ROT1 + GRUEN3, 24 SEK.
2060 DATA 17,30          :REM ROT1 + GELB3, 3 SEK.
```

Restliche Teile des Programms wie in der Aufgabenstellung.

**A5c:** Füllen Sie die Leerstellen in dem Ausgabeteil des folgenden SPS-Programms aus. Orientieren Sie sich dazu an dem Signal-Zeit-Diagramm der Aufgabe A5a. Die Zuordnung der logischen Verknüpfungen zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln entsprechend dem Signal-Zeit-Diagramm in Aufgabe A5a durchführen.

```
; AUFGABE A5C
; STEUERUNG DES FAHRZEUGVERKEHRS

; INITIALISIEREN DER ZEITGEBER          (siehe Seite 55)

; ABLAUFKETTE (GESTEUERT DURCH ZEITGEBER) (siehe Seite 55)

; AUSGABESTEuerung

U  M 00      ;   IN PHASE 1
O  M 03      ;   UND PHASE 4
O  M 04      ;   UND PHASE 5
O  M 05      ;   UND PHASE 6
=  A 00      ;8  ROT1 SCHALTEN

U  M 00      ;   IN PHASE 1
O  M 02      ;   UND PHASE 3
=  A 01      ;9  GELB1 SCHALTEN

U  M 01      ;   IN PHASE 2
=  A 02      ;10 GRUEN1 SCHALTEN

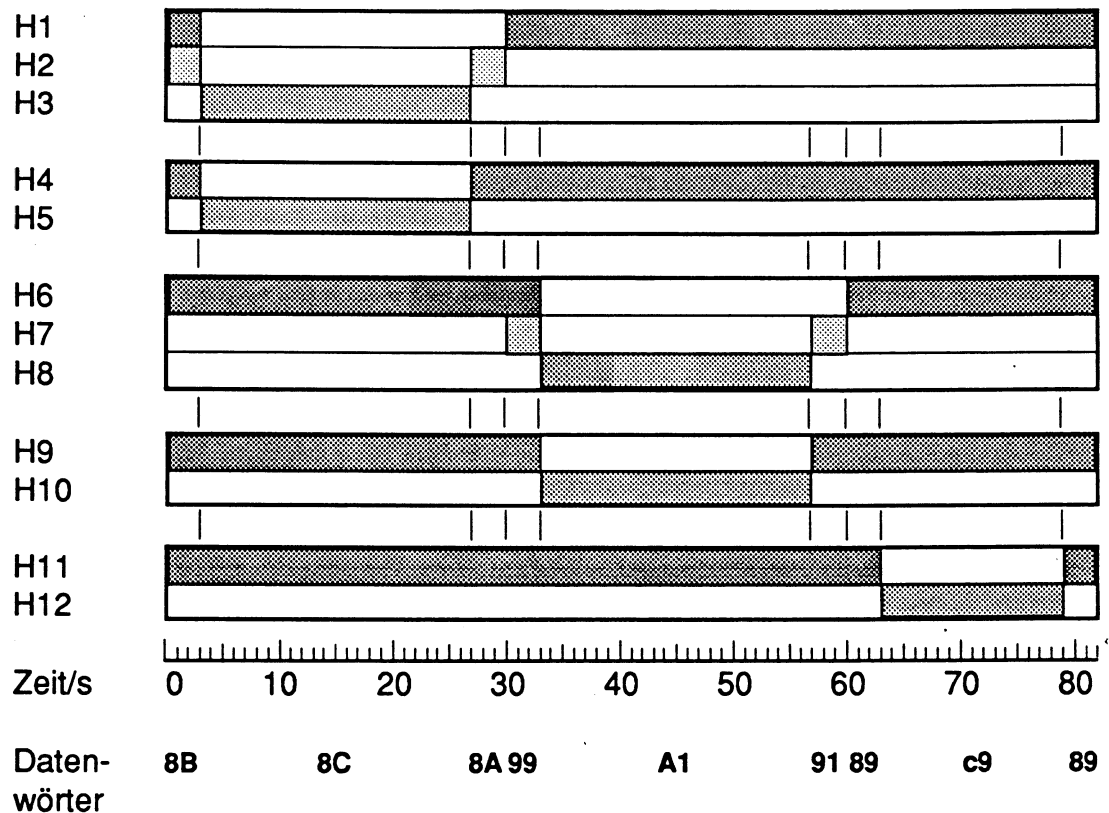
U  M 00      ;   IN PHASE 1
O  M 01      ;   UND PHASE 2
O  M 02      ;   UND PHASE 3
O  M 03      ;   UND PHASE 4
=  A 03      ;11 ROT3 SCHALTEN

U  M 03      ;   IN PHASE 4
O  M 05      ;   UND PHASE 6
=  A 04      ;12 GELB3 SCHALTEN

U  M 04      ;   IN PHASE 5
=  A 05      ;13 GRUEN3 SCHALTEN

END
```

**A6a:** Füllen Sie folgendes Signal-Zeit-Diagramm aus. Schraffieren Sie (vorzugsweise mit Farbstiften) die Zeitabschnitte, in denen die Lampen leuchten. Beginnen Sie mit der Rot-Gelb-Phase der Verkehrsampel 1. Schreiben Sie die Datenwörter, die zu den jeweiligen Abschnitten gehören, unter das Signal-Zeit-Diagramm.



**A6b:** Geben Sie die Datenwörter der Reihenfolge nach mit Hilfe des Out-Kommandos des MAT 85 aus. Kontrollieren Sie die Richtigkeit der Ampelsteuerung.

**A6c:** Füllen Sie die Leerzeilen in dem folgenden Assemblerprogramm mit Befehlen aus. Benutzen Sie den LXI-Befehl zum Laden der festgelegten Zeitdauern (Zwei-Byte-Konstanten) und den MVI-Befehl zum Laden der erforderlichen Datenwörter (Ein-Byte-Konstanten). Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln und Fußgängerampeln entsprechend dem Signal-Zeit-Diagramm aus Aufgabe A6a durchführen.

```
E03C 01 1E00      PHASE7: LXI      B,001E    ;3 SEKUNDEN
E03F 3E 89        MVI      A,89      ;ROT1 + ROT3
E041 D3 00        OUT      AUSGABE
E043 CD 5DE0      CALL     DELAY
E046 01 A000      PHASE8: LXI      B,00A0    ;16 SEKUNDEN
E049 3E C9        MVI      A,0C9     ;ROT1 + ROT3 + GRUEN5
E04B D3 00        OUT      AUSGABE
E04D CD 5DE0      CALL     DELAY
E050 01 1E00      PHASE9: LXI      B,001E    ;3 SEKUNDEN
E053 3E 89        MVI      A,89     ;ROT1 + ROT3
E055 D3 00        OUT      AUSGABE
E057 CD 5DE0      CALL     DELAY
E05A              ;
E05A C3 00E0      JMP      PHASE1    ;ENDLOSSCHLEIFE
```

Restliche Teile des Programms wie in der Aufgabenstellung.

- ▲ **A6d:** Füllen Sie die Leerstellen in dem Datenbereich des folgenden Assemblerprogramms durch Konstanten aus. Die Konstanten nach den DW-Assembleranweisungen sind die festgelegten Zeitdauern (Zwei-Byte-Konstanten) und die Konstanten nach den DB-Assembleranweisungen die zuvor ermittelten Datenwörter. Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrs- und Fußgängerampeln entsprechend dem Signal-Zeit-Diagramm aus Aufgabe A6a durchführen.

```
E02D          ; TABELLE DER AMPELPHASEN
E02D 8B          TAB:  DB      8B          ;ROT1 + GELB1 + ROT3
E02E 1E00        DW      001E          ;3 SEKUNDEN
E030 8C          DB      8C          ;GRUEN1 + ROT3
E031 F000        DW      00F0          ;24 SEKUNDEN
E033 8A          DB      8A          ;GELB1 + ROT3
E034 1E00        DW      001E          ;3 SEKUNDEN
E036 99          DB      99          ;ROT1 + ROT3 + GELB3
E037 1E00        DW      001E          ;3 SEKUNDEN
E039 A1          DB      0A1          ;ROT1 + GRUEN3
E03A F000        DW      00F0          ;24 SEKUNDEN
E03C 91          DB      91          ;ROT1 + GELB3
E03D 1E00        DW      001E          ;3 SEKUNDEN
E03F 89          DB      89          ;ROT1 + ROT3
E040 1E00        DW      001E          ;3 SEKUNDEN
E042 C9          DB      0C9          ;ROT1 + ROT3 + GRUEN5
E043 A000        DW      00A0          ;16 SEKUNDEN
E045 89          DB      89          ;ROT1 + ROT3
E046 1E00        DW      001E          ;3 SEKUNDEN
E048          ;
```

Restliche Teile des Programms wie in der Aufgabenstellung.

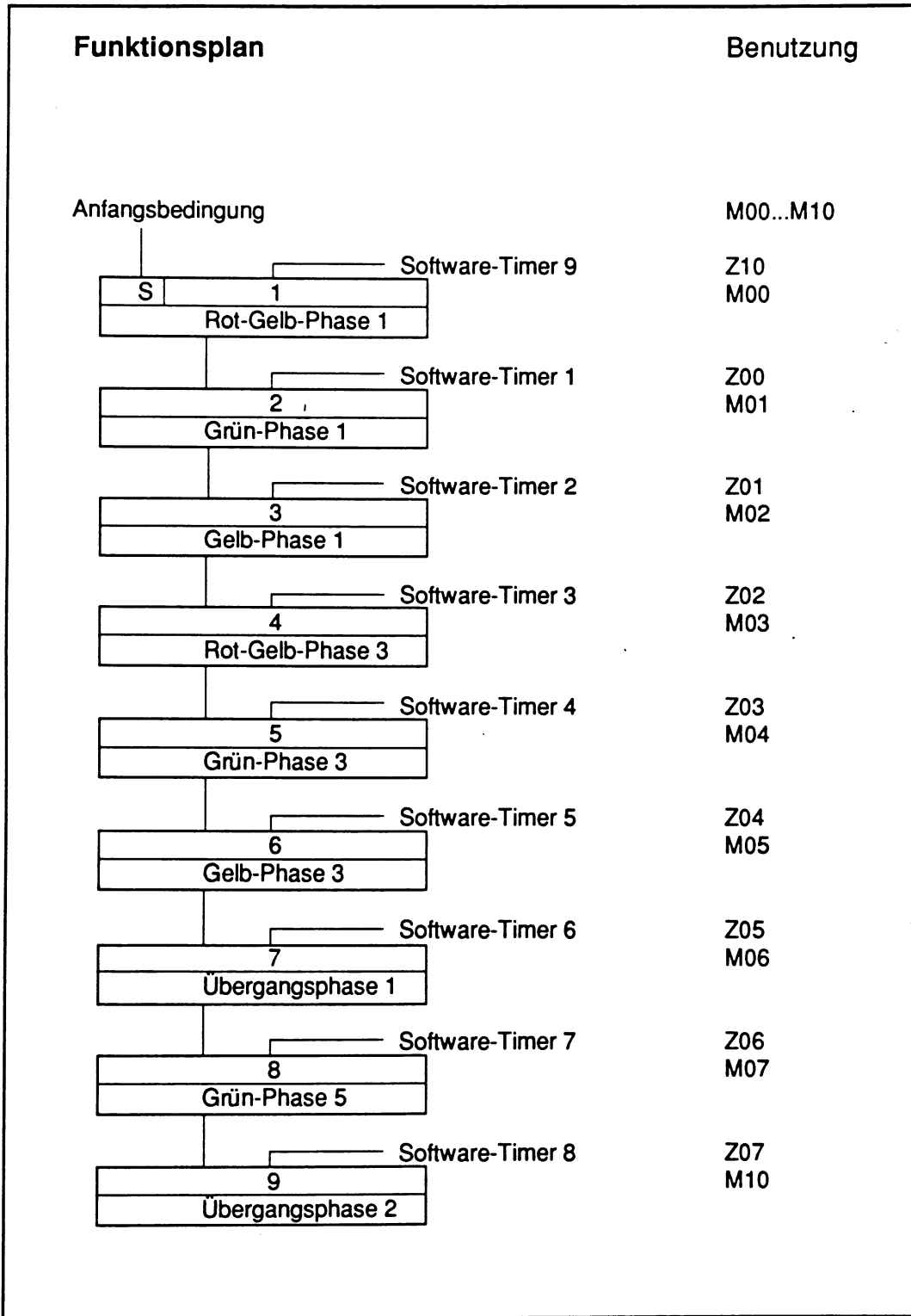
**A6c:** Füllen Sie die Leerstellen in den Befehlen des folgenden BASIC-Programms mit Befehlen aus. Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln und Fußgängerampeln entsprechend dem Signal-Zeit-Diagramm aus Aufgabe A6a durchführen.

```
10 REM AUFGABE A6C
20 REM STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
30 REM
40 A=0                                :REM 8-BIT-PARALLEL-AUSGABE
100 Z=30                             :REM 3 SEKUNDEN
110 OUT A,139                        :REM ROT1 + GELB1 + ROT3
120 GOSUB 1000
200 Z=240                            :REM 24 SEKUNDEN
210 OUT A,140                        :REM GRUEN1 + ROT3
220 GOSUB 1000
300 Z=30                             :REM 3 SEKUNDEN
310 OUT A,138                        :REM GELB1 + ROT3
320 GOSUB 1000
400 Z=30                             :REM 3 SEKUNDEN
410 OUT A,153                        :REM ROT1 + ROT3 + GELB3
420 GOSUB 1000
500 Z=240                            :REM 24 SEKUNDEN
510 OUT A,161                        :REM ROT1 + GRUEN3
520 GOSUB 1000
600 Z=30                             :REM 3 SEKUNDEN
610 OUT A,145                        :REM ROT1 + GELB3
620 GOSUB 1000
700 Z=30                             :REM 3 SEKUNDEN
710 OUT A,137                        :REM ROT1 + ROT3
720 GOSUB 1000
800 Z=160                            :REM 16 SEKUNDEN
810 OUT A,201                        :REM ROT1 + ROT3 + GRUEN5
820 GOSUB 1000
900 Z=30                             :REM 3 SEKUNDEN
910 OUT A,137                        :REM ROT1 + ROT3
920 GOSUB 1000
999 GOTO 100
1000 REM VERZOEGERUNG (Z = ZEIT IN 1/10 SEKUNDEN)
1010 FOR I=1 TO Z
1020 FOR J=1 TO 98
1030 NEXT J
1040 NEXT I
1050 RETURN
```

- ▲ **A6d:** Füllen Sie die Leerstellen in den DATA-Zeilen des folgenden BASIC-Programms durch Konstanten aus. Die erste Konstante in einer DATA-Zeile ist jeweils eines der zuvor ermittelten Datenwörter. Die zweite Konstante in einer Zeile ist die dazugehörige Phasendauer in 1/10-Sekunden. Die Zuordnung der Konstanten zu den Phasen entnehmen Sie den Kommentaren. Das Programm soll die Steuerung der Verkehrsampeln entsprechend dem Signal-Zeit-Diagramm aus Aufgabe A6a durchführen.

```
10 REM AUFGABE A6D
20 REM STEUERUNG DES FAHRZEUG- UND FUSSGAENGERVERKEHRS
30 REM VARIANTE MIT DEM TABELLENVERFAHREN
40 REM
50 A=0                      :REM 8-BIT-PARALLEL-AUSGABE
100 P=9                     :REM ZAHL DER PHASEN
110 RESTORE 2010            :REM ZEIGER AUF TABELLENANFANG
120 READ B,Z                :REM AUSGABEBYTE UND ZEITDAUER
130 OUT A,B                 :REM AUSGABE
140 GOSUB 1000
150 P=P-1                   :REM PHASENNUMMER
160 IF P>0 THEN GOTO 120
170 GOTO 100                :REM GESAMTPROZESS WIEDERHOLEN
1000 REM VERZOEGERUNG (Z = ZEIT IN 1/10 SEKUNDEN)
1010 FOR I=1 TO Z
1020 FOR J=1 TO 98
1030 NEXT J
1040 NEXT I
1050 RETURN
2000 REM DATENBEREICH
2010 DATA 139,30           :REM ROT1 + GELB1 + ROT3, 3 SEK.
2020 DATA 140,240         :REM GRUEN1 + ROT3, 24 SEK.
2030 DATA 138,30          :REM GELB1 + ROT3, 3 SEK.
2040 DATA 153,30          :REM ROT1 + ROT3 + GELB3, 3 SEK.
2050 DATA 161,240         :REM ROT1 + GRUEN3, 24 SEK.
2060 DATA 145,30          :REM ROT1 + GELB3, 3 SEK.
2070 DATA 137,30          :REM ROT1 + ROT3, 3 SEK.
2080 DATA 201,160         :REM ROT1 + ROT3 + GRUEN5, 16 SEK.
2090 DATA 137,30          :REM ROT1 + ROT3, 3 SEK.
```

**A6c:** Zeichnen Sie den Funktionsplan der Ablaufkette. Geben Sie auch die benutzten Schrittmerker und Software-Timer an.





**A6d:** Füllen Sie die Leerstellen in dem Ausgabeteil des folgenden SPS-Programms aus. Orientieren Sie sich dazu an dem Signal-Zeit-Diagramm der Aufgabe A6a. Die Zuordnung der logischen Verknüpfungen zu den Phasen entnehmen Sie den Kommentaren.

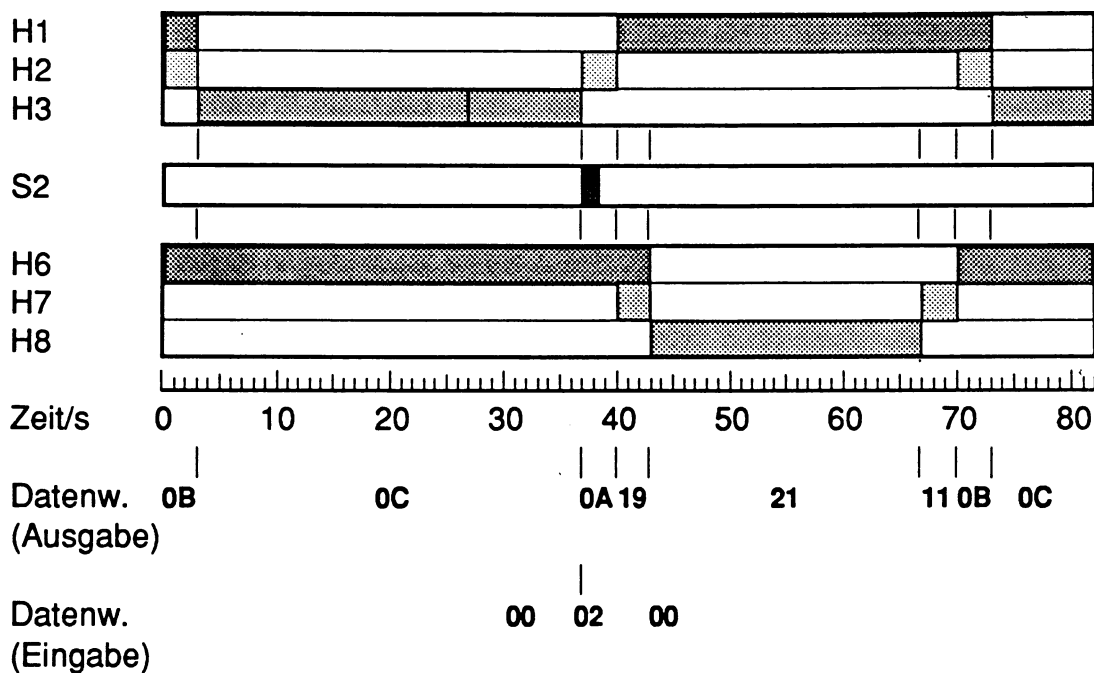
```
; AUSGABESTEuerung
U  M 00      ; IN  PHASE 1
O  M 03      ; UND PHASE 4
O  M 04      ; UND PHASE 5
O  M 05      ; UND PHASE 6
O  M 06      ; UND PHASE 7
O  M 07      ; UND PHASE 8
O  M 10      ; UND PHASE 9
=  A 00      ;11 ROT1 SCHALTEN
U  M 00      ; IN  PHASE 1
O  M 02      ; UND PHASE 3
=  A 01      ;12 GELB1 SCHALTEN
U  M 01      ; IN  PHASE 2
=  A 02      ;13 GRUEN1 SCHALTEN
U  M 00      ; IN  PHASE 1
O  M 01      ; UND PHASE 2
O  M 02      ; UND PHASE 3
O  M 03      ; UND PHASE 4
O  M 06      ; UND PHASE 7
O  M 07      ; UND PHASE 8
O  M 10      ; UND PHASE 9
=  A 03      ;14 ROT3 SCHALTEN
U  M 03      ; IN  PHASE 4
O  M 05      ; UND PHASE 6
=  A 04      ;15 GELB3 SCHALTEN
U  M 04      ; IN  PHASE 5
=  A 05      ;16 GRUEN3 SCHALTEN
U  M 07      ; IN  PHASE 8
=  A 06      ;17 GRUEN5 SCHALTEN
```

Restliche Teile des Programms wie in der Aufgabenstellung.

**A7a:** Füllen Sie folgendes Signal-Zeit-Diagramm aus. Schraffieren Sie (vorzugsweise mit Farbstiften) die Zeitabschnitte in denen die Lampen leuchten. Beginnen Sie mit der Rot-Gelb-Phase der Verkehrsampel 1. Nehmen Sie an, daß zehn Sekunden nach Ablauf der zeitgeführten Grün-Phase der Verkehrsampel 1 die "Induktionsschleife" S2 betätigt wird.

Schreiben Sie die Datenwörter, die zu den jeweiligen Abschnitten gehören, unter das Signal-Zeit-Diagramm. Beachten Sie, daß die Fußgängerampeln dunkel bleiben sollen.

Schreiben Sie auch die Datenwörter auf, die von der 8-Bit-Parallel-Eingabe eingelesen werden: vor, nach und während der Betätigung von S2. Gehen Sie dabei davon aus, daß keine anderen Melde-Einrichtungen betätigt werden und die Schalter von E03 bis E07 Low-Pegel erzeugen.



Das Programm der Aufgaben A7b-d ist in der Aufgabenstellung abgedruckt.

Beantworten Sie die folgenden Fragen. Versuchen Sie jeweils zunächst, die Antwort aus dem Verständnis des Programms zu geben. Überprüfen Sie anschließend Ihre Antwort im Experiment.

**A7b:** Das Programm ist gestartet, die Verkehrsampel 1 zeigt grün an. Warten Sie ggf. noch 24 Sekunden. Drücken Sie dann kurz die Taste S2. Was geschieht?

**Die Hauptstraße wird sofort gesperrt. Es läuft ein vollständiger Zyklus der Ampelphasen ab, um das Fahrzeug von der Nebenstraße einbiegen zu lassen.**

**A7c:** Drücken Sie so lange die Taste S2, bis die Gelb-Phase der Verkehrsampel 1 erscheint. Lassen Sie S2 los. Drücken Sie S2 erneut während der Gelb-Phase der Verkehrsampel 3 und halten Sie S2 fest. Wie lange dauert die Grün-Phase der Verkehrsampel 1?

**Die Grün-Phase dauert 24 Sekunden an.**

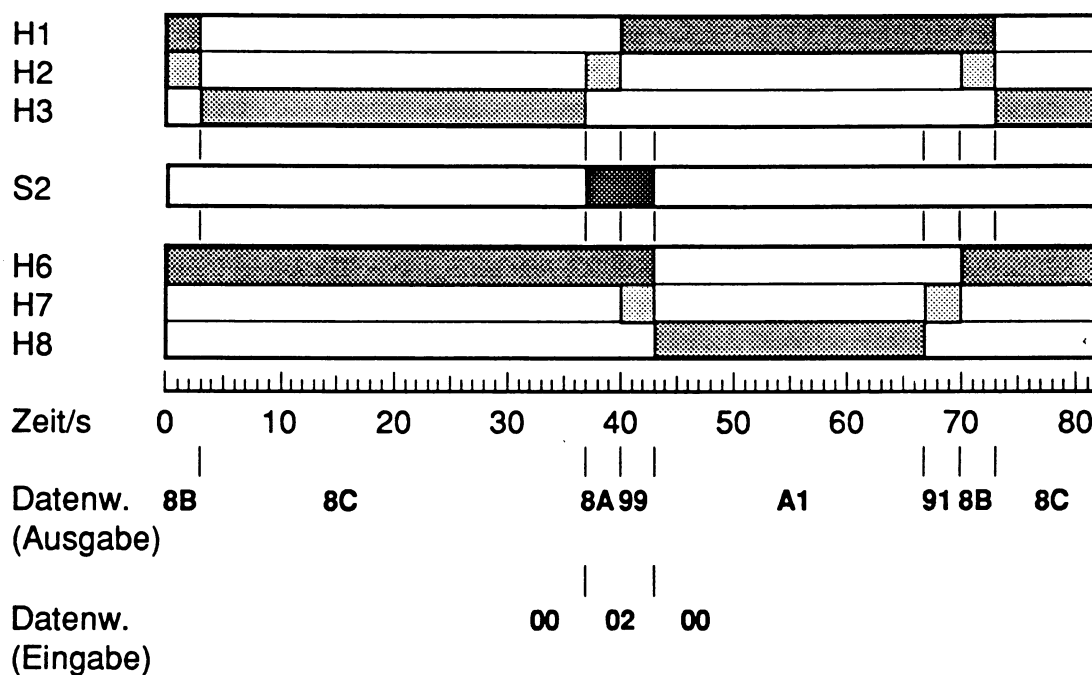
**A7d:** Starten Sie das Programm erneut. Gleich nach Erscheinen der Grün-Phase der Verkehrsampel 1 drücken Sie kurz S2. Was geschieht?

**Die Grün-Phase dauert weiterhin an, denn S2 wird nur im prozeßgeführten Teil der Grün-Phase (nach 24 Sekunden) abgefragt. Der Impuls geht verloren, d.h. hat keine Auswirkung.**

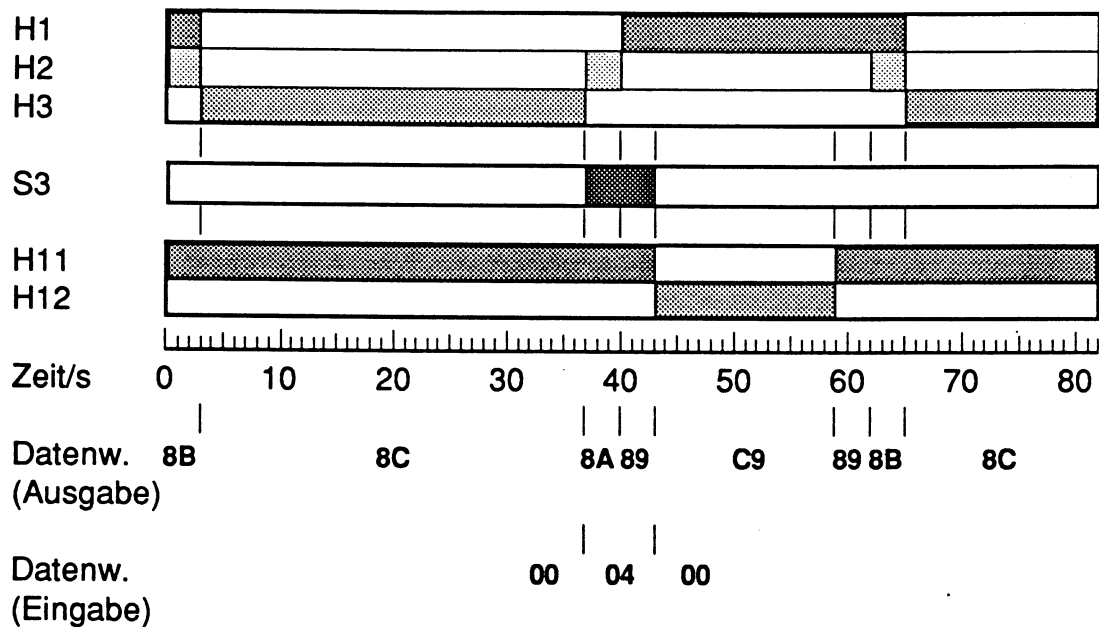
**A8a:** Füllen Sie folgendes Signal-Zeit-Diagramm aus. Schraffieren Sie (vorzugsweise mit Farbstiften) die Zeitabschnitte, in denen die Lampen leuchten. Beginnen Sie mit der Rot-Gelb-Phase der Verkehrsampel 1. Nehmen Sie an, daß zehn Sekunden nach Ablauf der zeitgeführten Grün-Phase der Verkehrsampel 1 die "Induktionsschleife" S2 betätigt wird.

Schreiben Sie die Datenwörter, die zu den jeweiligen Abschnitten gehören, unter das Signal-Zeit-Diagramm. Beachten Sie, daß die Fußgängerampeln in Betrieb sind.

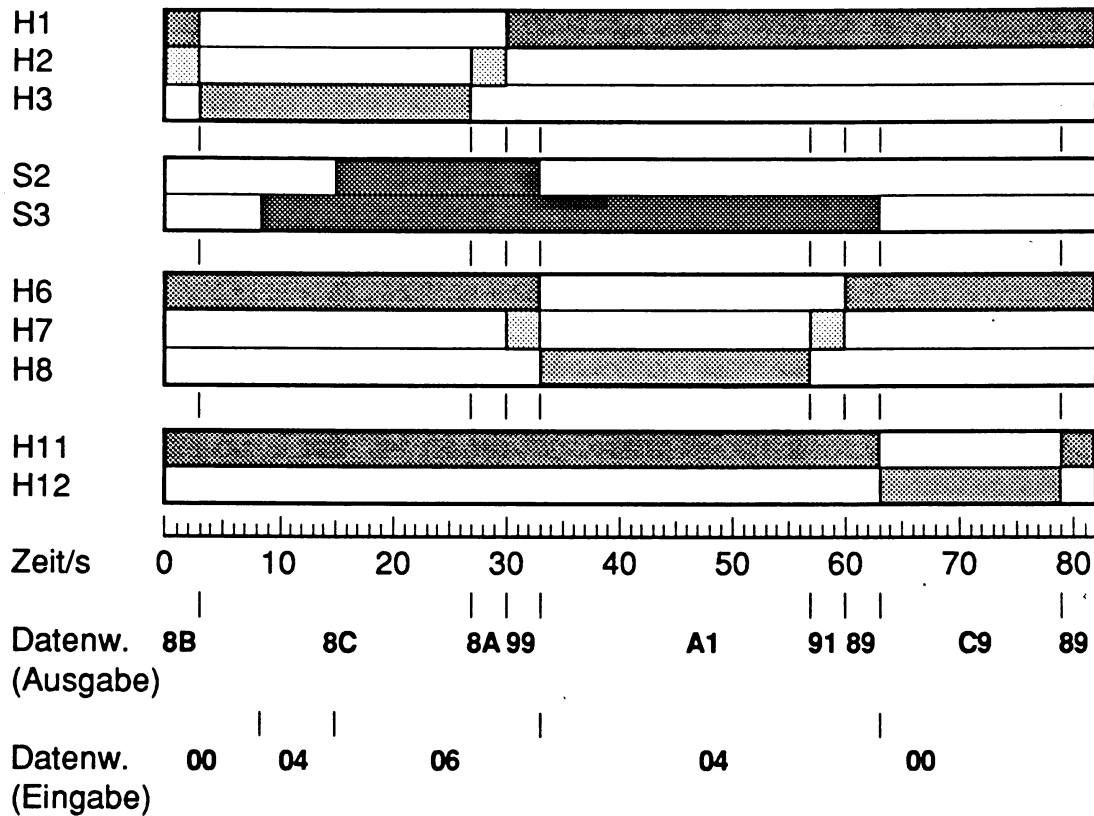
Schreiben Sie auch die Datenwörter auf, die von der 8-Bit-Parallel-Eingabe eingelesen werden: vor, nach und während der Betätigung von S2. Gehen Sie dabei davon aus, daß keine anderen Melde-Einrichtungen betätigt werden und die Schalter von E03 bis E07 Low-Pegel erzeugen.



**A8b:** Füllen Sie wie in Aufgabe A8a folgendes Signal-Zeit-Diagramm aus. Beginnen Sie mit der Rot-Gelb-Phase der Verkehrsampel 1. Nehmen Sie an, daß zehn Sekunden nach Ablauf der zeitgeführten Grün-Phase der Verkehrsampel 1 die Meldetaste S3 betätigt wird.



**A8c:** Füllen Sie wie in Aufgabe A8a folgendes Signal-Zeit-Diagramm aus. Beginnen Sie mit der Rot-Gelb-Phase der Verkehrsampel 1. Nehmen Sie an, daß noch während der zeitgeführten Grün-Phase der Verkehrsampel 1 die "Induktionsschleife" S2 und die Meldetaste S3 betätigt und festgehalten werden. Die Tasten werden jeweils wieder losgelassen, wenn für den entsprechenden Verkehrsweg "Grün" gemeldet wird.



Das Programm von Aufgabe A8d-A8f ist in der jeweiligen Aufgabenstellung abgedruckt.

Beantworten Sie die folgenden Fragen. Versuchen Sie jeweils zunächst, die Antwort aus dem Verständnis des Programms zu geben. Überprüfen Sie anschließend Ihre Antwort im Experiment.

**A8d:** Das Programm ist gestartet, die Verkehrsampel 1 zeigt grün an. Warten Sie ggf. noch 24 Sekunden. Drücken Sie dann kurz die Taste S3. Was geschieht?

**Die Hauptstraße wird sofort gesperrt. Es läuft ein folgender Zyklus der Ampelphasen ab, um das Fahrzeug von der Nebenstraße einbiegen zu lassen:**

**Verkehrsampel 1: gelb, rot, rot-gelb, grün.**

**Verkehrsampel 3:** rot-gelb, grün, gelb, rot

**A8e:** Starten Sie das Programm erneut. Gleich nach Erscheinen der Grün-Phase der Verkehrsampel 1 drücken Sie kurz S2. Was geschieht?

**Die Grün-Phase dauert weiterhin an, und es findet kein Umschalten statt, denn S2 wird nur im prozeßgeführten Teil der Grün-Phase (nach 24 Sekunden) abgefragt. Der Impuls hatte keine Auswirkung.**

**A8f:** Starten Sie das Programm erneut. Gleich nach Erscheinen der Grün-Phase der Verkehrsampel Sie S2 und S3. Lassen Sie die Tasten erst los, wenn die Grün-Phase der jeweiligen Ampel erscheint, also das Fahrzeug fahren bzw. der Fußgänger gehen kann. Was geschieht?

**Die Hauptstraße wird nach Ablauf der Mindestdauer gesperrt (Verkehrsampel 1: gelb, rot), anschließend wird die Nebenstraße freigegeben (Verkehrsampel 3: rot-gelb, grün, gelb) und zum Schluß der Fußgängerübergang (Übergangsphase 1, grün, Übergangsphase 2). Nach Ablauf dieser beiden Ampelzyklen wird die Hauptstraße wieder freigegeben (Verkehrsampel 1: rot-gelb, grün)**

**A8g (nur SPS):** Starten Sie das Programm erneut. Gleich nach Erscheinen der Grün-Phase der Verkehrsampel 1 drücken Sie die Taste S2. Lassen Sie die Taste innerhalb der Gelb-Phase der Verkehrsampel 1 los. Was geschieht?

**Die Ampelsteuerung bleibt in der Gelb-Phase hängen.**

**Erläuterung:** Durch das Drücken einer der beiden Tasten S2 oder S3 (hier S2) wird der Übergang von der Grün-Phase zur Gelb-Phase bewirkt. Für die anschließende Programmverzweigung entweder zur Rot-Gelb-Phase der Verkehrsampel 3 oder der Übergangsphase 1 der Fußgängerampel 5 fehlt jedoch das entsprechende Signal.

**A9a:** Stellen Sie hier Ihre Kritik an dem Programm aus Aufgabe A8 zusammen. Gegebenenfalls laden Sie nochmals das Programm der Aufgabe A8 und führen Sie die Tests gemäß Aufgabe A8d bis A8f (bzw. A8g) sowie eigene Tests durch.

**Die Steuerung der Ampelanlage durch das Programm der Aufgabe 8 weist folgende Mängel auf:**

1. Die Melde-Einrichtung muß so lange betätigt werden, bis der entsprechende Verkehrsweg grün erhält. Andernfalls ist nicht gewährleistet, daß der Verkehrsweg auch tatsächlich berücksichtigt wird.
2. Im Falle des SPS-Programms kann das Programm gar hängen bleiben, wenn das Signal der Melde-Einrichtung während der Gelb-Phase der Verkehrsampel 1 wieder zurückfällt.



**A9b:** Das nachstehende Assemblerprogramm steuert die Ampeln abhängig vom Verkehrsaufkommen der Nebenstraße und vom Fußgängerverkehr. Impulse von der "Induktionsschleife" S2 und der Meldetaste S3 werden gespeichert.

Das Programm ist eine Weiterentwicklung desjenigen aus Aufgabe A8. Füllen Sie die Leerstellen in dem Programm aus. Beachten Sie, daß das Registerpaar HL auf die Speicherzelle TASTEN zeigt, diese also unter Verwendung des Pseudoregister M verwendet werden kann.

```

E047 01 1E00      PHASE6: LXI      B,001E  ;3 SEKUNDEN
E04A 3E 91        MVI      A,91      ;ROT1 + GELB3
E04C D3 00        OUT      AUSGABE
E04E CD 80E0      CALL     DELAY
E051 7E          MOV      A,M        ;SPEICHER ZURUECKSETZEN
E052 E6 FD        ANI      OFD       ;BIT 1 LOESCHEN
E054 77          MOV      M,A        ;WIEDER SPEICHERN
E055 7E          PHASE7:
...
E06F 01 1E00      PHASE9: LXI      B,001E  ;3 SEKUNDEN
E072 3E 89        MVI      A,89      ;ROT1 + ROT3
E074 D3 00        OUT      AUSGABE
E076 CD 80E0      CALL     DELAY
E079 7E          MOV      A,M        ;SPEICHER ZURUECKSETZEN
E07A E6 FB        ANI      OFB       ;BIT 2 LOESCHEN
E07C 77          MOV      M,A        ;WIEDER SPEICHERN
E07D              ;
E07D C3 03E0      JMP      PHASE1    ;ENDLOSSCHLEIFE
E080              ;
E080              ;      UNTERPROGRAMM DELAY
E080 C5          DELAY: PUSH     B      ;REGISTER RETTEN
E081 D5          PUSH     D
E082 F5          PUSH     PSW
E083 11 5C11      DELAY1: LXI      D,115C  ;ZEITBASIS 1/10 S
E086 DB 00        DELAY2: IN       EINGABE ;TASTENSTELLUNGEN EINLESEN
E088 B6          ORA      M          ;IM DATENWORT SPEICHERND
E089 77          MOV      M,A        ;ABLEGEN
E08A 1B          DCX      D          ;GEGEN NULL ZAEHLEN
E08B 7A          MOV      A,D        ;TESTE AUF NULL
E08C B3          ORA      E
E08D C2 86E0      JNZ      DELAY2    ;SCHLEIFENENDE DELAY2
E090 0B          DCX      B          ;GEGEN NULL ZAEHLEN
E091 78          MOV      A,B        ;TESTE AUF NULL
E092 B1          ORA      C
E093 C2 83E0      JNZ      DELAY1    ;SCHLEIFENENDE DELAY1
E096 F1          POP      PSW
E097 D1          POP      D
E098 C1          POP      B
E099 C9          RET                ;UNTERPROGRAMM DELAY
E09A              ;
E09A              ; DATENSPEICHER
E09A 00          TASTEN: DB      0      ;SPEICHER FUER EINGABEBYTE

```

**A9c:** Prüfen Sie das Programm, indem Sie zu verschiedenen Zeitpunkten die Melde-Einrichtungen S2 bzw. S3 betätigen.

Erstellen Sie ein Protokoll Ihrer Tests.

**Das Protokoll sollte als wesentliches Element erkennen lassen, daß durch die Speicherung der Impulse der Melde-Einrichtungen nun ein befriedigendes Verhalten der Steuerung erreicht wird.**

**A9b:** Das nachstehende Assemblerprogramm steuert die Ampeln abhängig vom Verkehrsaufkommen der Nebenstraße und vom Fußgängerverkehr. Impulse von der "Induktionsschleife" S2 und der Meldetaste S3 werden gespeichert.

Das Programm ist eine Weiterentwicklung desjenigen aus Aufgabe A8. Füllen Sie die Leerstellen im Programm aus.

```
600 Z=30                      :REM 3 SEKUNDEN
610 OUT A,145                  :REM ROT1 + GELB3
620 GOSUB 1000
630 T=T AND 253                :REM BIT 1 LOESCHEN
700 IF (T AND 4) = 0 THEN GOTO 100
710 Z=30                      :REM 3 SEKUNDEN
720 OUT A,137                  :REM ROT1 + ROT3
730 GOSUB 1000
800 Z=160                     :REM 16 SEKUNDEN
810 OUT A,201                  :REM ROT1 + ROT3 + GRUEN5
820 GOSUB 1000
900 Z=30                      :REM 3 SEKUNDEN
910 OUT A,137                  :REM ROT1 + ROT3
920 GOSUB 1000
930 T=T AND 251                :REM BIT 2 LOESCHEN
999 GOTO 100
1000 REM VERZOEGERUNG (Z = ZEIT IN 1/10 SEKUNDE)
1010 REM UND SPEICHERN VON TASTENBETAETIGUNGEN
1020 FOR I=1 TO Z
1030 FOR J=1 TO 7
1040 T=T OR INP (E)            :REM TASTEN LESEN UND SPEICHERN
1050 NEXT J
1060 NEXT I
1070 RETURN
```

**A9c:** Prüfen Sie das Programm, indem Sie zu verschiedenen Zeitpunkten die Melde-Einrichtungen S2 bzw. S3 betätigen.

Erstellen Sie ein Protokoll Ihrer Tests.

Das Protokoll sollte als wesentliches Element erkennen lassen, daß durch die Speicherung der Impulse der Melde-Einrichtungen nun ein befriedigendes Verhalten der Steuerung erreicht wird.

**A9b:** Das nachstehende Assemblerprogramm steuert die Ampeln abhängig vom Verkehrsaufkommen der Nebenstraße und vom Fußgängerverkehr. Impulse von der "Induktionsschleife" S2 und der Meldetaste S3 werden in den Merkern M21 bzw. M22 gespeichert.

Das Programm ist eine Weiterentwicklung desjenigen aus Aufgabe A8. Füllen Sie die Leerstellen im Programm aus.

```
U   Z 05           ;   WENN PHASE 6 ABGELAUFEN
=R  M 21           ; 9 RUECKSETZE IMPULSSPEICHER DER IND.SCHL. 3

U   Z 10           ;   WENN PHASE 9 ABGELAUFEN
=R  M 22           ; 14 RUECKSETZE IMPULSSPEICHER TASTE 5

; SPEICHERUNG DER EINGABEIMPULSE

U   E 01           ;   IMPULS DER INDUKTIONSSCHLEIFE 3
=S  M 21           ; 16 IN MERKER 21 SPEICHERN

U   E 02           ;   IMPULS DER FUSSGAENGERTASTE 5
=S  M 22           ; 17 IN MERKER 22 SPEICHERN
```

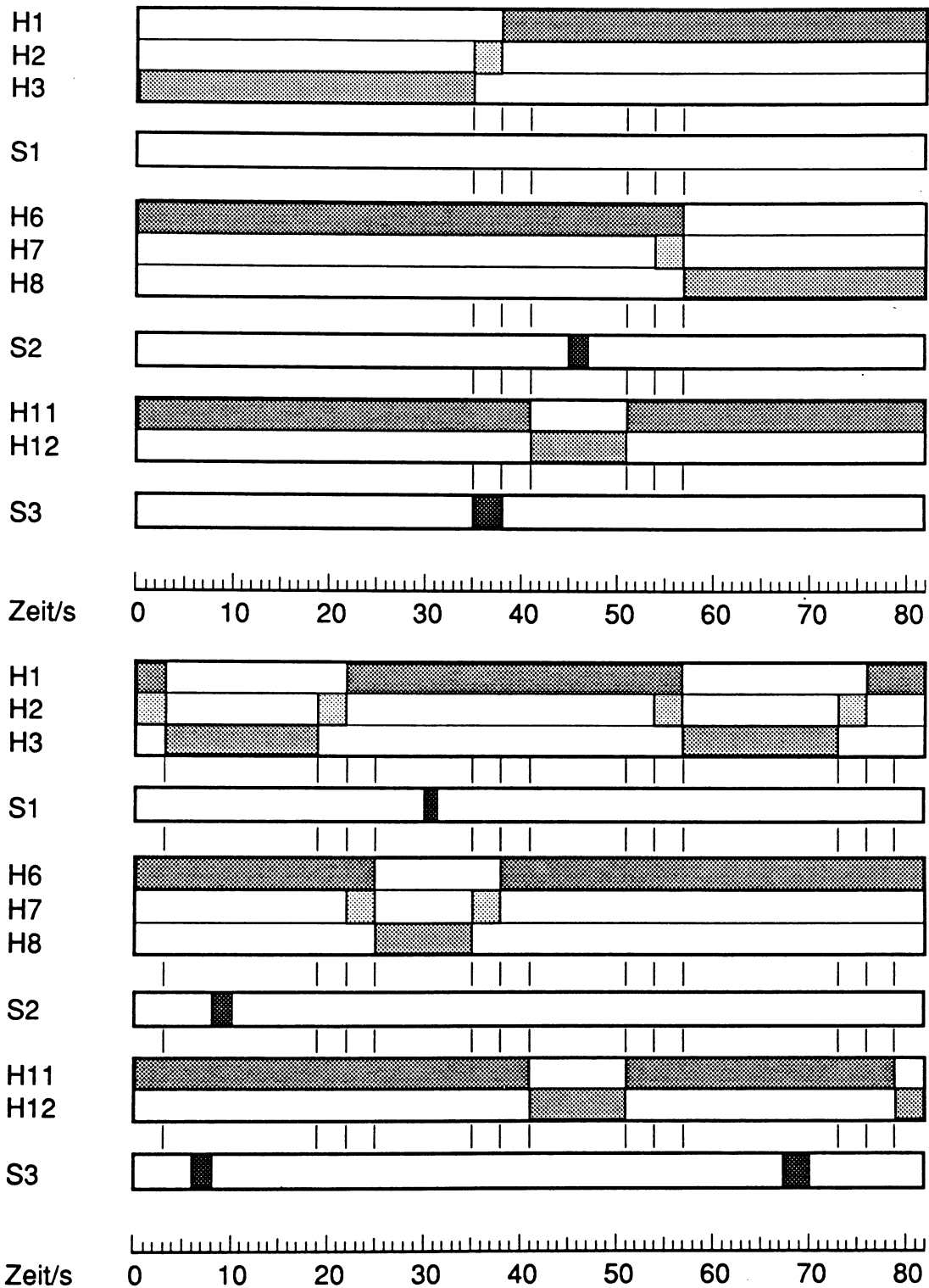
**A9c:** Prüfen Sie das Programm, indem Sie zu verschiedenen Zeitpunkten die Melde-Einrichtungen S2 bzw. S3 betätigen.

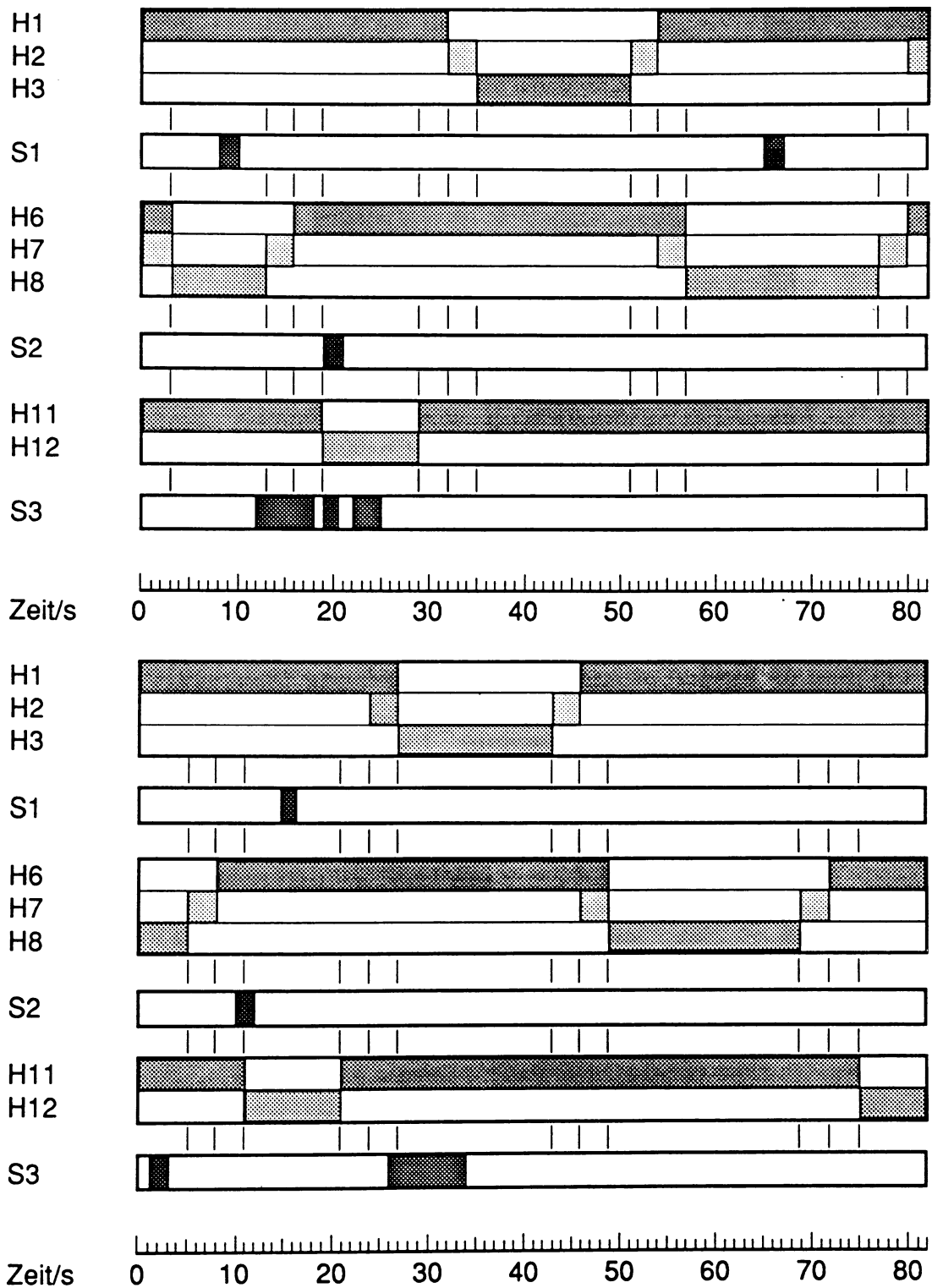
Erstellen Sie ein Protokoll Ihrer Tests.

**Das Protokoll sollte als wesentliches Element erkennen lassen, daß durch die Speicherung der Impulse der Melde-Einrichtungen nun ein befriedigendes Verhalten der Steuerung erreicht wird.**

**Zudem sollte das Hängen des Programms in der Gelb-Phase 1 (s. Aufgabe A8g) überprüft worden sein und nicht mehr auftreten.**

**A10a:** Vervollständigen Sie folgende Signal-Zeit-Diagramme entsprechend der Funktionsbeschreibung der Ampelsteuerung:





Die Programme der Aufgabe A10 sind in *Kapitel 4* abgedruckt. Prüfen Sie die korrekte Funktion des jeweiligen Programms. Legen Sie sich selbst die geeigneten Prüfmethoden zurecht und beschreiben Sie sie für die Verkehrsampel 1 in Stichpunkten. Im einzelnen sollten Sie prüfen:

**A10b:** Einhaltung der Zeiten und der korrekten Ampelschaltungen bei normaler Verkehrsdichte.

**A10c:** Verdopplung der Mindestdauer der Grün-Phase bei hohem Verkehrsaufkommen.

**A10d:** Korrekte Reaktion auf kein Verkehrsaufkommen bei Verkehrsampel 3 oder Fußgängerampel 5 oder beiden.

**A10e:** Berücksichtigung der Ampeln im Uhrzeigersinn.

▲ **A10f:** Testen Sie weitere selbst gewählte Situationen.

**Bei der Auswahl der Tests ist auf folgende Punkte zu achten:**

**Wenn die Dauer von Grün-Phasen ausgemessen werden soll, so muß spätestens innerhalb der Mindestdauer der Grün-Phase bereits wieder eine Taste eines anderen Verkehrswegs gedrückt worden sein, da sonst die zeitgeführte Grün-Phase unbemerkt in die (im Prinzip beliebig lang dauernde) prozeßgeführte Grün-Phase übergeht.**

**Assembler und BASIC-Programme:**

**Die Programme starten immer so, daß im ersten Durchlauf durch die Ampelphasen die Mindestdauer der hohen Verkehrsdichte gewählt wird. Deshalb sollte zunächst durch Signale an den Tasten S1 bis S3 jeder Verkehrsweg einmal berücksichtigt worden sein, bevor z.B. mit Zeitmessungen begonnen wird.**

**BASIC-Programm:**

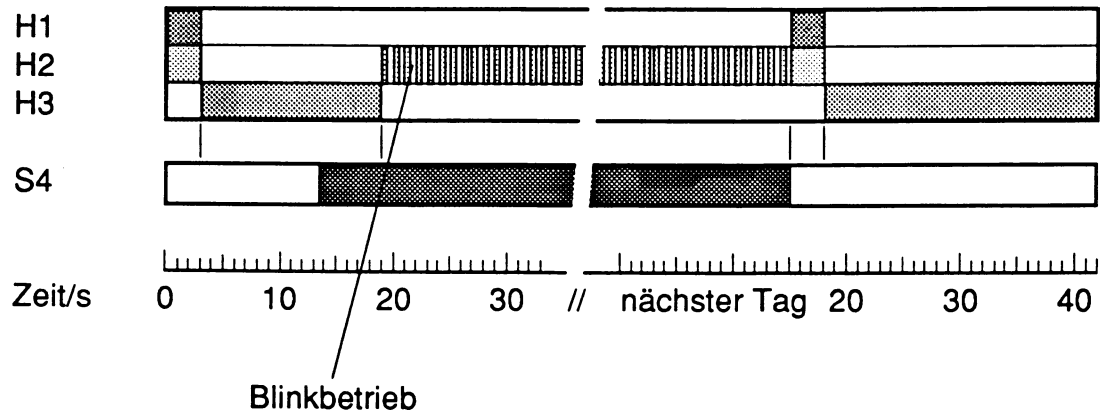
**Beachten Sie, daß beim BFZ-Steuer-BASIC-Programm (nicht CP/M) der Zeittakt 1/10 Sekunde nicht stimmt und daher alle Zeiten um einen bestimmten (stets gleichen) Faktor länger sind.**

▲ **A10g (nur Assembler und BASIC):** Wann läuft die Uhr zum ersten Mal über, und wie wäre dem abzuhelpfen?

**Die Uhr läuft nach 32767 Zehntel-Sekunden = ca. 55 Minuten über.**

**Durch Speicherung in Drei-Byte- oder längeren Worten oder getrennt nach Zehntel-Sekunden, Minuten und Stunden wird die Uhr verbessert.**

**A11a:** Ergänzen Sie das nachstehende Signal-Zeit-Diagramm





**A11b:** Das nachstehende Assemblerprogramm führt die Steuerung der Ampeln sowohl im Tag- als auch im Nachtbetrieb durch.

Das Programm ist eine Weiterentwicklung desjenigen aus Aufgabe A10. Es enthält bereits die korrekten Übergangsbedingungen vom Tag- zum Nachtbetrieb. Der Nachtbetrieb und der Übergang zum Tagbetrieb ist noch nicht programmiert. Füllen Sie die Leerstellen im Programm aus.

```
E12F          ; NACHT-BETRIEB
E12F 01 0500  PHAS10: LXI      B,0005 ;1/2 SEKUNDE
E132 3E 12      MVI      A,12  ;GELB1 + GELB3
E134 D3 00      OUT      AUSGABE
E136 CD 50E1    CALL     DELAY
E139 01 0500  PHAS11: LXI      B,0005 ;1/2 SEKUNDE
E13C 3E 00      MVI      A,00  ;ALLE LAMPEN AUS
E13E D3 00      OUT      AUSGABE
E140 CD 50E1    CALL     DELAY
E143          ;
E143          ; UEBERPRUEFEN, OB TAG-SCHALTUNG
E143 DB 00      IN       EINGABE ;EINGAENGE LESEN
E145 E6 08      ANI      08     ;SCHALTUHR MASKIEREN
E147 C2 2FE1    JNZ      PHAS10 ;WEITER NACHT-BETRIEB
E14A CD C7E1    CALL     INIT   ;DATENBEREICH LOESCHEN
E14D C3 0EE0    JMP      START  ;NUN TAG-BETRIEB
```

Rest des Programms wie in der Aufgabenstellung.

**A11c:** Prüfen Sie die Funktionstüchtigkeit des Programms, indem Sie in verschiedenen Situationen zwischen Tag- und Nachtbetrieb hin- und herschalten. Protokollieren Sie Ihre Tests.

**Die Tests des Programms sollten insbesondere folgende Punkte des Pflichtenhefts (siehe *Abschnitt 4.11*) prüfen:**

**Ist das Umschalten auf den Nachtbetrieb tatsächlich zum Ende jeder Grün-Phase möglich?**

**Bleibt die Mindestdauer der Grün-Phasen gewahrt?**

**Beginnt nach dem Zurückschalten zum Tagbetrieb der Ampelzyklus mit der Grün-Phase der Hauptstraße?**

**Sind Anforderungen, die noch beim Umschalten in den Nachtbetrieb unerledigt blieben, beim Umschalten in den Tagbetrieb gelöscht worden?**

**A11b:** Das nachstehende BASIC-Programm führt die Steuerung der Ampeln sowohl im Tag- als auch im Nachtbetrieb durch.

Das Programm ist eine Weiterentwicklung desjenigen aus Aufgabe A10. Es enthält bereits die korrekten Übergangsbedingungen vom Tag- zum Nachtbetrieb. Der Nachtbetrieb und der Übergang zum Tagbetrieb ist noch nicht programmiert. Füllen Sie die Leerstellen im Programm aus.

```
2000 REM NACHT-BETRIEB
2010 Z=5                                :REM 1/2 SEKUNDE
2020 OUT A,18                          :REM GELB1 + GELB3
2030 GOSUB 1000
2040 Z=5                                :REM 1/2 SEKUNDE
2050 OUT A,0                            :REM ALLE LAMPEN AUS
2060 GOSUB 1000
2070 IF (INP(E) AND 8)=8 THEN GOTO 2000
2080 GOTO 50                            :REM WEITER IM TAG-BETRIEB
```

Rest des Programms wie in der Aufgabenstellung.

**A11c:** Prüfen Sie die Funktionstüchtigkeit des Programms, indem Sie in verschiedenen Situationen zwischen Tag- und Nachtbetrieb hin- und herschalten. Protokollieren Sie Ihre Tests.

**Die Tests des Programms sollten insbesondere folgende Punkte des Pflichtenhefts (siehe *Abschnitt 4.11*) prüfen:**

**Ist das Umschalten auf den Nachtbetrieb tatsächlich zum Ende jeder Grün-Phase möglich?**

**Bleibt die Mindestdauer der Grün-Phasen gewahrt?**

**Beginnt nach dem Zurückschalten zum Tagbetrieb der Ampelzyklus mit der Grün-Phase der Hauptstraße?**

**Sind Anforderungen, die noch beim Umschalten in den Nachtbetrieb unerledigt blieben, beim Umschalten in den Tagbetrieb gelöscht worden?**

**A11b:** Das nachstehende SPS-Programm führt die Steuerung der Ampeln sowohl im Tag- als auch im Nachtbetrieb durch.

Das Programm ist eine Weiterentwicklung desjenigen aus Aufgabe A10. Es enthält bereits eine korrekte Übergangsbedingung vom Tag- zum Nachtbetrieb. Der Nachtbetrieb und der Übergang zum Tagbetrieb ist noch nicht programmiert. Füllen Sie die Leerstellen im Programm aus.

; TAG/NACHT-SCHALTUNG

```
U   E 03      ;   WENN NACHTZEIT
U   Z 01      ;   UND PHASE 2 ABGELAUFEN
=R  Z 01      ;
=R  M 01      ;
=R  A 07      ;   FUSSGAENGERAMPELN ABSCHALTEN
=S  Z 11      ;   STARTE BLINKPHASE GELB
=S  M 11      ;27

U   E 03      ;   WENN NACHTZEIT
U   Z 04      ;   UND PHASE 5 ABGELAUFEN
=R  Z 04      ;
=R  M 04      ;
=R  A 07      ;   FUSSGAENGERAMPELN ABSCHALTEN
=S  Z 11      ;   STARTE BLINKPHASE GELB
=S  M 11      ;28

U   E 03      ;   WENN NACHTZEIT
U   Z 07      ;   UND PHASE 8 ABGELAUFEN
=R  Z 07      ;
=R  M 07      ;
=R  A 07      ;   FUSSGAENGERAMPELN ABSCHALTEN
=S  Z 11      ;   STARTE BLINKPHASE GELB
=S  M 11      ;29

U   Z 11      ;   WENN BLINKPHASE GELB ABGELAUFEN
=R  Z 11      ;
=R  M 11      ;
=S  Z 12      ;   STARTE PAUSENPHASE GELB
=S  M 12      ;30

U   E 03      ;   WENN NACHTZEIT
U   Z 12      ;   UND PAUSENPHASE GELB ABGELAUFEN
=R  Z 12      ;
=R  M 12      ;
=S  Z 11      ;   STARTE BLINKPHASE GELB
=S  M 11      ;31
```

```
UN  E 03      ;   WENN TAGESZEIT
U   Z 12      ;   UND PAUSENPHASE GELB ABGELAUFEN
=R  Z 12      ;
=R  M 12      ;
=S  A 07      ;   FUSSGAENGERAMPELN EINSCHALTEN
=R  M 20      ;   SPEICHER DER MELDEIMPULSE LOESCHEN
=R  M 21      ;
=R  M 22      ;
=S  Z 00      ;   STARTE PHASE 1
=S  M 00      ;32
```

Rest des Programms wie in der Aufgabenstellung.

**A11c:** Prüfen Sie die Funktionstüchtigkeit des Programms, indem Sie in verschiedenen Situationen zwischen Tag- und Nachtbetrieb hin- und herschalten. Protokollieren Sie Ihre Tests.

**Die Tests des Programms sollten insbesondere folgende Punkte des Pflichtenhefts (siehe *Abschnitt 4.11*) prüfen:**

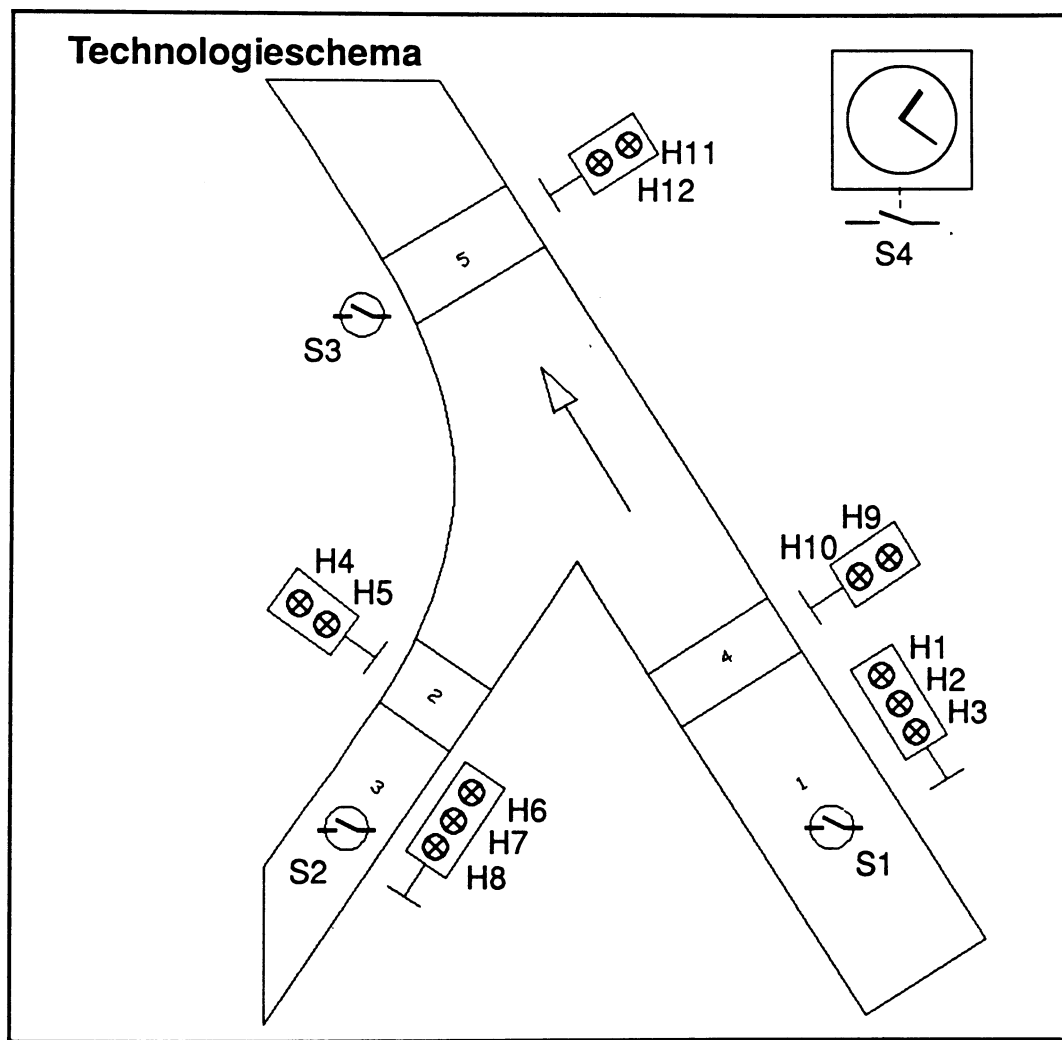
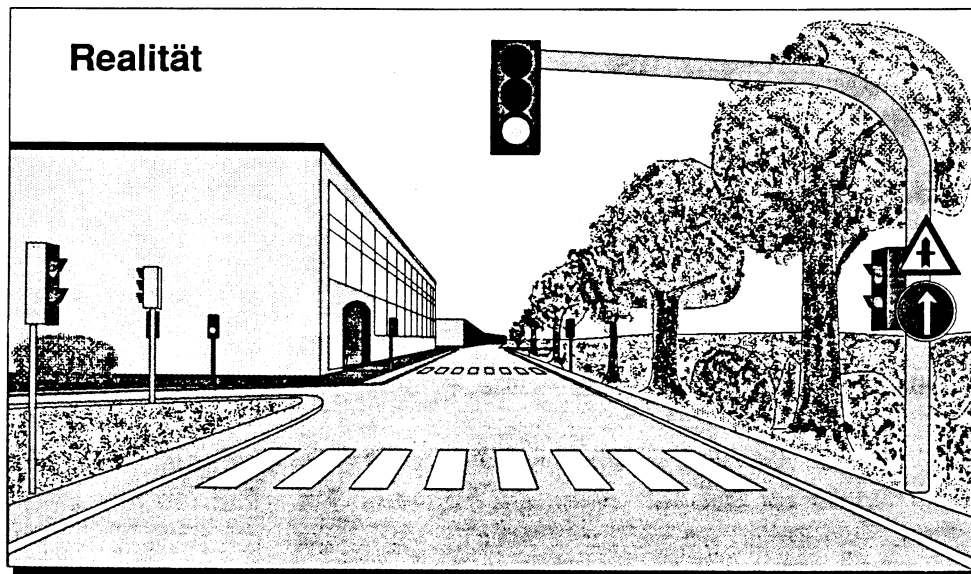
**Ist das Umschalten auf den Nachtbetrieb tatsächlich zum Ende jeder Grün-Phase möglich?**

**Bleibt die Mindestdauer der Grün-Phasen gewahrt?**

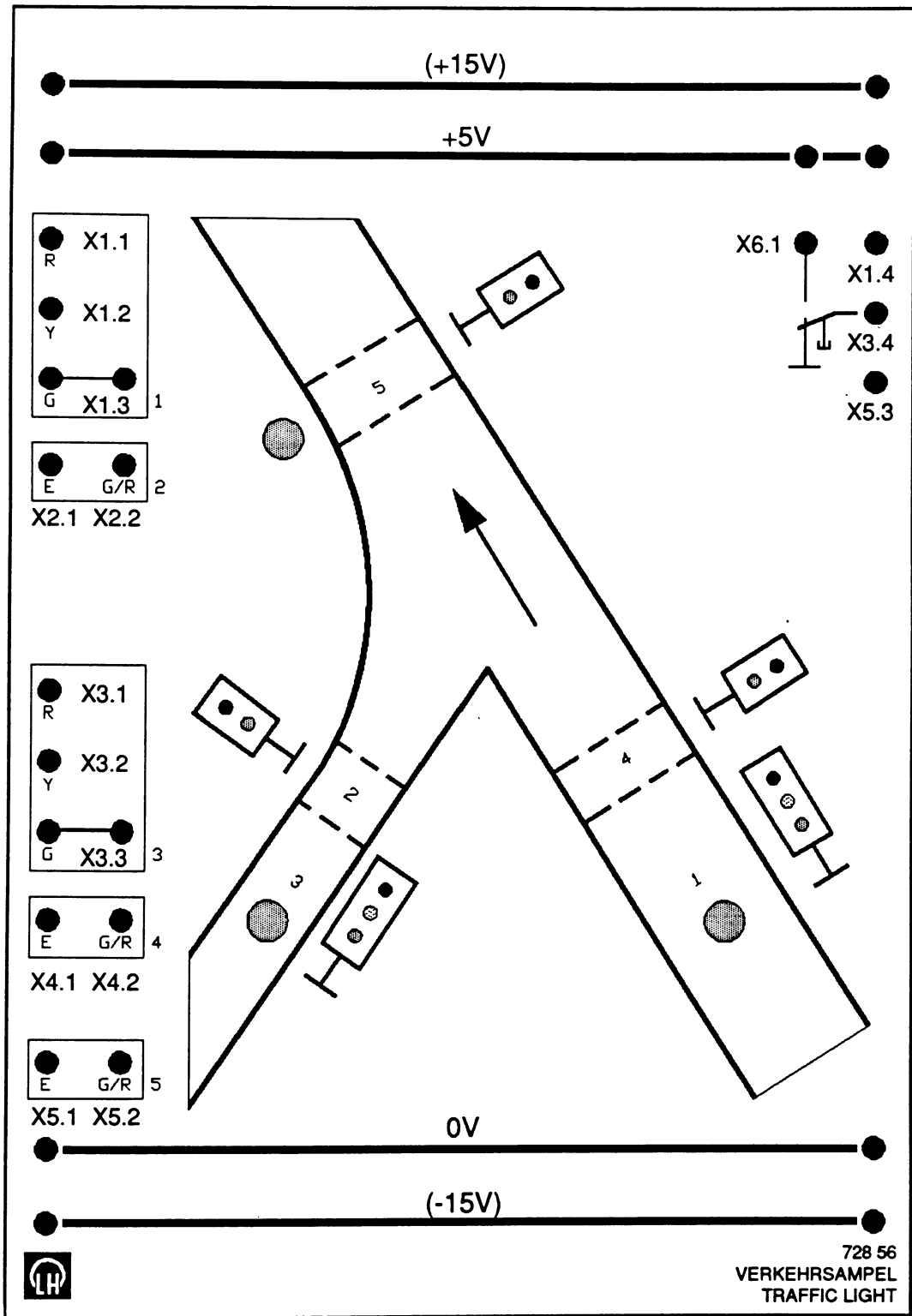
**Beginnt nach dem Zurückschalten zum Tagbetrieb der Ampelzyklus mit der Grün-Phase der Hauptstraße?**

**Sind Anforderungen, die noch beim Umschalten in den Nachtbetrieb unerledigt blieben, beim Umschalten in den Tagbetrieb gelöscht worden?**

## 6. Kopiervorlagen



Ansicht der Experimentierplatte VERKEHRSAMPEL

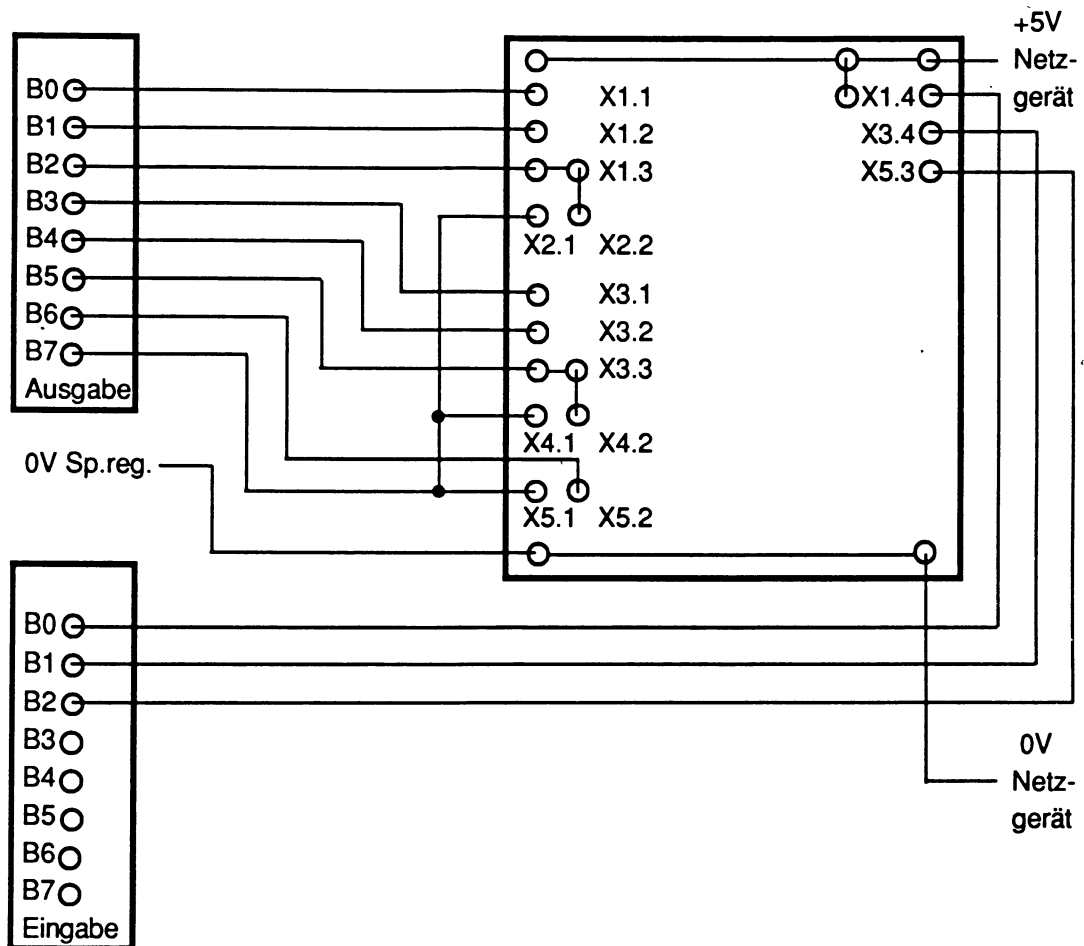


## 6. Kopiervorlagen

### Zuordnungsliste

Kenn- zeichen	Ein-/ Ausgang (Computer)	Klemme (Modell)	Betriebs- mittel	Funktion
S1	E00	X1.4	Indukt. schleife	Ankunft eines Fahrzeugs an Ampel 1
S2	E01	X3.4	Indukt. schleife	Ankunft eines Fahrzeugs an Ampel 3
S3	E02	X5.3	Taster	Anforderung Überweg 5
S4	E03	--	Schalter	Zeitschaltuhr Tag / Nacht
H1	A00	X1.1	Lampe	Anzeige "rot" Ampel 1
H2	A01	X1.2	Lampe	Anzeige "gelb" Ampel 1
H3	A02	X1.3	Lampe	Anzeige "grün" Ampel 1
H4 / H5	A07	X2.1	Signal	Betriebsfreigabe Ampel 2
"	A02	X2.2		"rot"/"grün" Ampel 2
H6	A03	X3.1	Lampe	Anzeige "rot" Ampel 3
H7	A04	X3.2	Lampe	Anzeige "gelb" Ampel 3
H8	A05	X3.3	Lampe	Anzeige "grün" Ampel 3
H9 / H10	A07	X4.1	Signal	Betriebsfreigabe Ampel 4
"	A05	X4.2		"rot"/"grün" Ampel 4
H11/ H12	A07	X5.1	Signal	Betriebsfreigabe Ampel 5
"	A06	X5.2		"rot"/"grün" Ampel 5

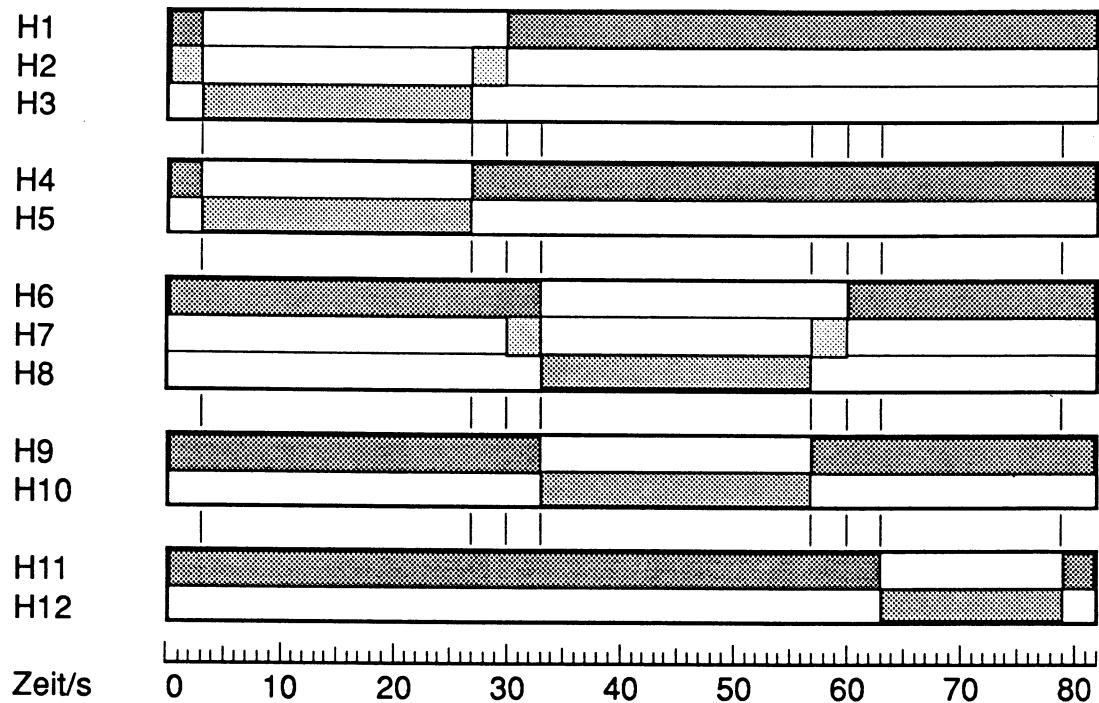
### Schaltbild der elektrischen Anschlüsse



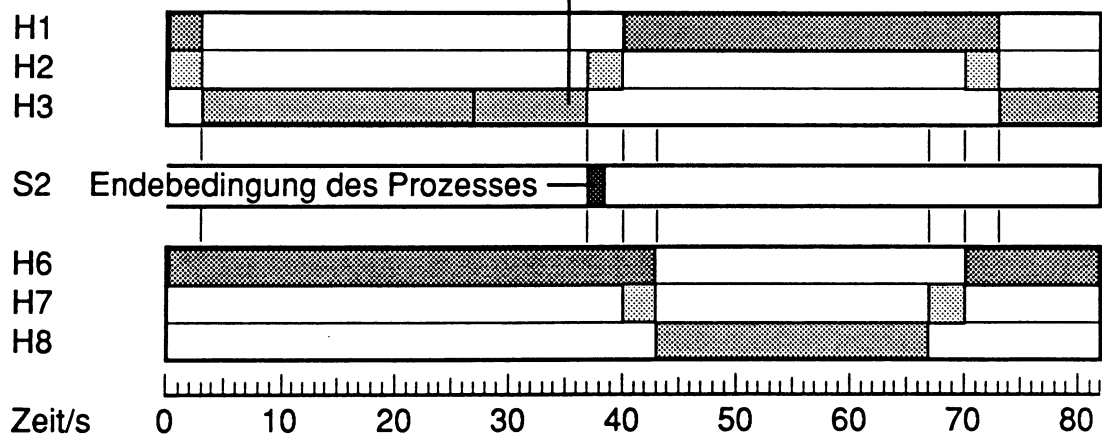


## 6. Kopiervorlagen

### Zeitgeführte Schritte...

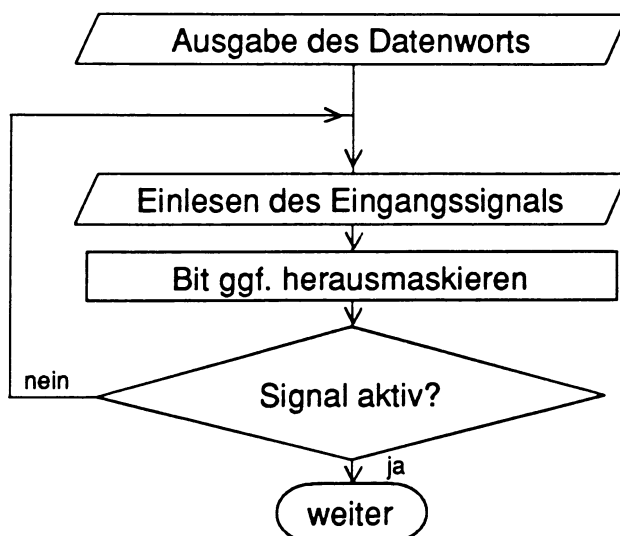
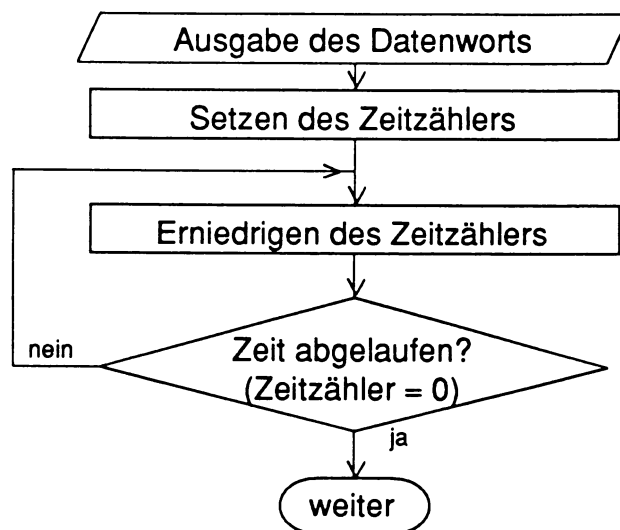


### ... und prozeßgeführter Schritt



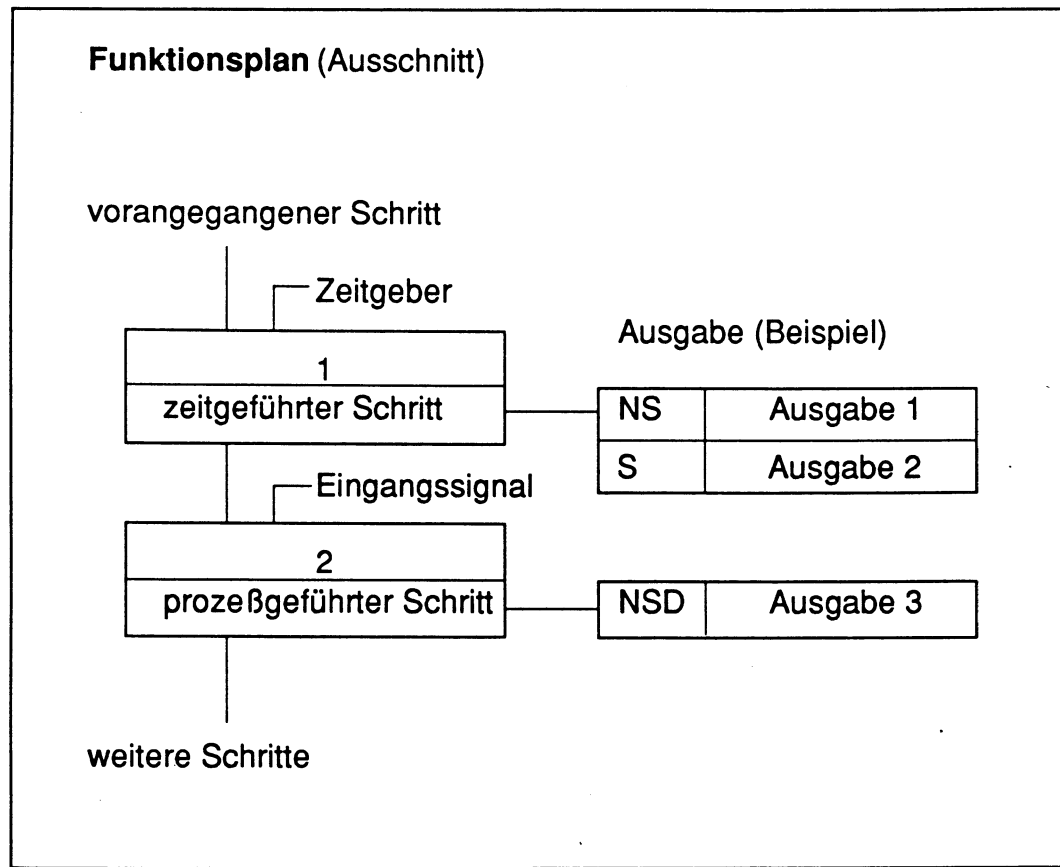
## Zeitgeführter und prozeßgeführter Schritt

## Programmablaufplan (Ausschnitt)



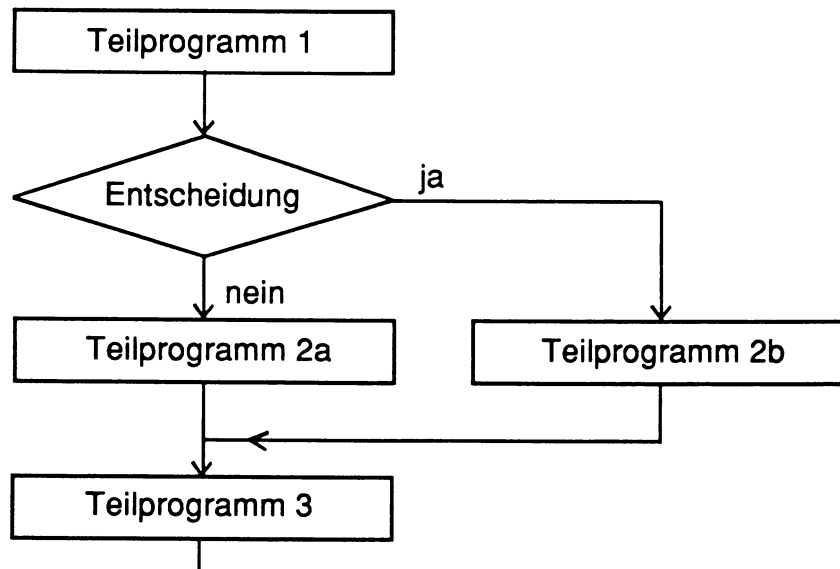
## 6. Kopiervorlagen

### Zeitgeführter und prozeßgeführter Schritt

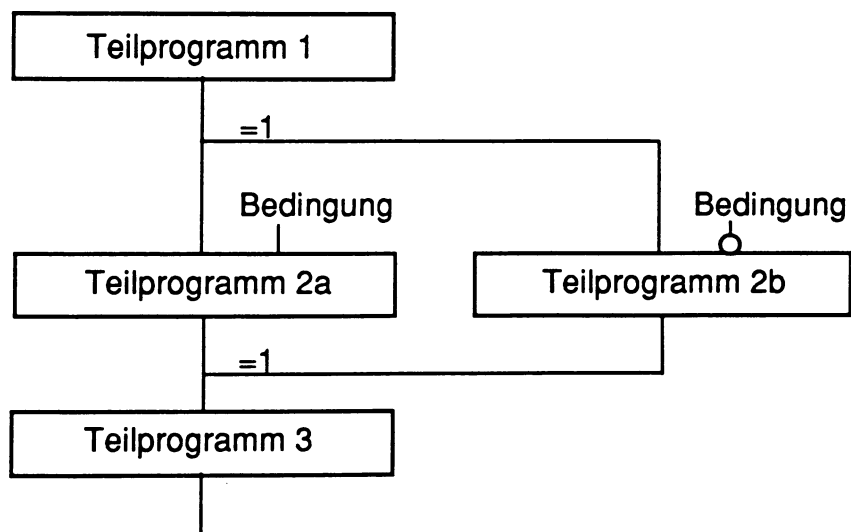


## Programmverzweigung

Programmablaufplan (schematisch)

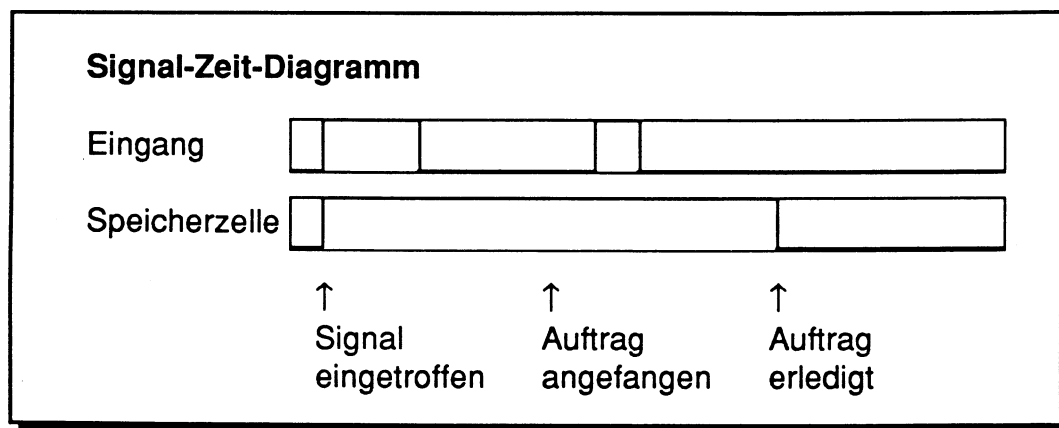


Funktionsplan (schematisch)



## 6. Kopiervorlagen

### Flankensteuerung



### Zahlendarstellung der zeitlichen Entwicklung

#### 1. Speichern

Eingang	Speicher- zelle	Ergebnis ODER	
00000000	00000000	00000000	
00000000	00000000	00000000	
00000010	00000000	00000010	S2 gedrückt
00000010	00000010	00000010	Impuls gespeichert
00000010	00000010	00000010	
00000000	00000010	00000010	S2 losgelassen
00000000	00000010	00000010	
00000000	00000010	00000010	
00000010	00000010	00000010	S2 nochmals gedrückt
00000010	00000010	00000010	keine Auswirkung
00000000	00000010	00000010	

↓ Zeit

#### 2. Löschen

Speicher- zelle	Maske	Ergebnis UND	
00110010	11111101	00110000	Auftrag erledigt

MFA-MEDIENSYSTEM

# Mikrocomputer- Technik

Handbuch zum  
Leybold-Ampelmodell

  
MEDIENSYSTEM

VGS