

PLANETH

**Ein kleincomputergesteuertes, interaktives,
audiovisuelles Lehrsystem,
speziell geeignet für den Einsatz
im naturwissenschaftlichen und technischen
Mittel- und Hochschulunterricht**

ABHANDLUNG

zur Erlangung
des Titels eines Doktors der Technischen Wissenschaften
der
**EIDGENÖSSISCHEN TECHNISCHEN HOCHSCHULE
ZÜRICH**

vorgelegt von

HEINRICH RUDOLF HÄBERLIN

dipl. El.-Ing. ETH
geboren am 9. Februar 1947
von Illighausen/TG

Angenommen auf Antrag von

Prof. Dr. G. Epprecht, Referent
Prof. Dr. H. Fischer, Korreferent

1978

Meinen Eltern gewidmet

Die vorliegende Arbeit wurde in den Jahren 1971 - 1978 am Mikrowellenlabor der Eidgenössischen Technischen Hochschule Zürich ausgeführt.

Dem Vorsteher des Instituts, Herrn Prof. Dr. G. Epprecht, möchte ich für sein lebhaftes Interesse an meiner Arbeit, seine vielen Anregungen und die grosszügige Unterstützung herzlich danken.

Mein besonderer Dank richtet sich auch an meinen Kollegen H. Müller, der in etwa 1½-jähriger Arbeit den SESAM-Compiler schrieb, ohne den die Herstellung der vielen Lehrprogramme für das Lehrsystem PLANETH äusserst mühsam gewesen wäre.

Dank schulde ich ferner Herrn U. Waber, der in den Jahren 1972 und 1973 zeitweise am Projekt mitarbeitete und der die heute noch verwendete Projektorsteuerung konzipierte. Auch Herrn H. Lindauer, der die Lehrkonsolen mechanisch zusammenbaute, und Herrn F. Amatore, der die Serienproduktion der Elektronik durchführte, möchte ich für ihren Einsatz bestens danken.

INHALTSVERZEICHNIS

ZUSAMMENFASSUNG	1
ABSTRACT	3
1. VON DER PROGRAMMIERTEN INSTRUKTION ZUM COMPUTERUNTERSTUETZTEN UNTERRICHT	5
1.1 Die programmierte Instruktion	6
1.2 Lehrautomaten	10
1.3 Der computerunterstützte Unterricht (CUU)	12
1.3.1 Kleincomputergesteuerte audiovisuelle Lehrsysteme	13
1.3.2 Grosscomputergesteuerte Lehrsysteme	14
1.3.3 Neuere Lehrstrategien bei grossrechnergesteuerten Lehrsystemen	16
1.4 Sinnvoller Einsatz des computerunterstützten Unterrichts	19
1.4.1 Die Kosten pro CUU-Stunde	20
1.4.2 Die Rentabilitätsgrenze neuer Lehrprogramme für bereits laufende Lehrsysteme	21
1.4.3 Gegenwärtige Einsatzmöglichkeiten für CUU	24
2. KONZEPT UND PRINZIPIELLE FUNKTIONSWEISE DES KLEINCOMPUTER- GESTEUERTEN LEHRSYSTEMS PLANETH	26
2.1 Kurzer historischer Rückblick	26
2.2 Konzept des Lehrsystems PLANETH	27
2.2.1 Diskussion des gewählten Konzeptes	30
2.3 Aufbau- und Funktionsprinzip von PLANETH	31
3. DIE PLANETH-LEHRKONSOLE	33
3.1 Die verschiedenen Möglichkeiten des Dialogs zwischen Konsole und Adressat	33
3.2 Das Prinzip der Datenübertragung zwischen Computer und Konsole	35
3.3 Gesamt-Blockschema und prinzipielle Funktionsweise der Konsole	36
3.4 Die Party-Line und die dazugehörigen Sende- und Empfangsschaltungen	40
3.4.1 Prinzipielle Schaltung der Party-Line	40

3.4.2	Die Ersatzschaltung der Party-Line und die Sende-/Empfangsschaltungen	42
3.4.3	Der zeitliche Verlauf der Party-Line-Signale	44
3.4.4	Der Einfluss des Uebersprechens	46
3.5	Die Konsolensteuereinheit (KS)	47
3.6	Dianummern-Speicher und Projektorsteuerungen	48
3.7	Lampensteuerung (LS)	51
3.8	Zeitsteuerung und Zeitanzeige (ZS)	53
3.9	Numerische Anzeige (NA)	55
3.10	Das PLANETH-Kassetten-Tonbandgerät	56
3.10.1	Wichtigste technische Daten des PLANETH- Kassettentonbandgeräts	60
3.11	Tonbandsteuerung	60
3.11.1	Das Konzept der Tonsequenzmarkierung	60
3.11.2	Der Ablauf der Suchvorgänge beim gewählten Markierungskonzept	64
3.11.3	Blockschema und Funktionsbeschreibung der Tonband-Steuerelektronik	65
3.12	Tastenspeicher (TA)	70
4.	DIE GERAETE DER ZENTRALEN STEUERUNG	74
4.1	Der Computer 2116B von Hewlett-Packard	74
4.2	Die am Computer angeschlossenen serienmässigen Ein- und Ausgabegeräte	75
4.2.1	Fernschreibmaschine	75
4.2.2	Lochstreifenleser	76
4.2.3	Lochstreifenstanzer	76
4.3	Das Party-Line-Vorschaltgerät	76
5.	DAS PLANETH - SYSTEMPROGRAMM	80
5.1	Grobflussdiagramm und prinzipielle Funktion des PLANETH-Systemprogramms	82
5.2	Abschätzung der Reaktionszeiten des PLANETH-Systemprogramms	85
5.2.1	Kurze Behandlung des Warteschlangenproblems mit einer Bedienungsstation	86
5.2.2	Die Reaktionszeit bei der Verarbeitung von Tasteninformatio	87
5.2.3	Warteschlangenprobleme beim Ausdrucken der Lehrprogramm-Schlussprotokolle	89
5.3	Die Konsolen-Status-Parameter	91

5.4	Das PLANETH-System-Hauptprogramm LEMAS	94
5.4.1	Wichtigste Aufgaben des Hauptprogramms LEMAS	94
5.4.2	Flussdiagramm des PLANETH-System-Hauptprogramms (LEMAS)	97
5.5	Das Unterprogramm LEBEV (Lehrprogramm-Befehls-Verarbeitung)	102
5.5.1	Die Bewertung der eingegebenen Antworten	110
5.6	Die Aufgaben der übrigen Unterprogramme	111
5.6.1	Das Unterprogramm LEDER: Datenverkehr mit Party-Line	111
5.6.2	Das Unterprogramm ZAHLD: Definition von Zahlen	112
5.6.3	Das Unterprogramm ZAHLA: Anzeige von Zahlen	112
5.6.4	Die Unterprogramme für die Textausgabe auf der Fernschreibmaschine: QIO4, .TTY. und LETEX	113
5.6.5	Das Unterprogramm BIMAP: Kompaktspeicherung von Daten und grösser-kleiner Vergleiche	115
6.	DIE HERSTELLUNG DER LEHRPROGRAMME	116
6.1	Probleme beim Einsatz von Lehrmaschinenprogrammen	116
6.2	Die Lehrstoffspeicher beim Lehrsystem PLANETH	117
6.3	Die für die Lehrprogrammherstellung nötigen Arbeiten	117
6.4	Der Aufwand für die Programmherstellung in Funktion der Komplexität des Lehrprogramms beim Lehrsystem PLANETH	119
6.5	Das Uebersetzungsprogramm (Compiler) für die Autorensprache SESAM	120
6.6	Die Autorensprache SESAM zum Lehrsystem PLANETH	121
6.6.1	Prinzipielle Eigenschaften und Befehlsvorrat	121
6.6.1.1	Aufbau eines Lehrprogrammbefehls	122
6.6.1.2	Uebersicht über die einzelnen Lehrprogramm- befehle	123
6.6.2	Die Deklarationsbefehle	126
6.6.2.1	Der PROG-Befehl	126
6.6.2.2	Der WERT-Befehl	128
6.6.2.3	Der UPRO-Befehl	129
6.6.2.4	Der ZIEL-Befehl	130
6.6.3	Die Befehle zur Ansteuerung der Konsolenperipherie- einheiten	130
6.6.3.1	Der BILD-Befehl	130
6.6.3.2	Der BAND-Befehl	132
6.6.3.3	Der LAMP-Befehl	134
6.6.3.4	Der ZEIG-Befehl	135

6.6.4	Die Befehle für die Antwortbewertung	136
6.6.4.1	Der BELA-Befehl	137
6.6.4.2	Der BEVA-Befehl	138
6.6.4.3	Der BENU-Befehl für die Fehlerklassen- einteilung und Bewertung von numerischen Konstruktivantworten	139
6.6.5	Die Befehle zur Steuerung des Programmablaufs	142
6.6.5.1	Der STOP-Befehl	142
6.6.5.2	Der GUMP-Befehl	143
6.6.5.3	Der WENN-Befehl	144
6.6.5.4	Der IMIN-Befehl	147
6.6.5.5	Der ZUFA-Befehl	148
6.6.5.6	Der RUFEBefehl	149
6.6.5.7	Der ENDE-Befehl	149
6.6.5.8	Der LEER-Befehl	150
6.6.5.9	Der .HALT-Befehl	150
6.6.6	Lehrprogrammbeispiele	151
6.6.6.1	Einfaches lineares Lehrprogramm ohne Ton	151
6.6.6.2	Einfaches verzweigtes Lehrprogramm ohne Ton	152
6.6.6.3	Vertontes Lehrprogramm mit Auswahl- und Konstruktivantworten	153
6.7	Die Herstellung der Diapositive	155
6.8	Die Tonaufnahmen	156
7.	DIE VOM PLANETH-SYSTEM GELIEFERTEN STATISTIKDATEN UND IHRE AUSWERTUNG	157
7.1	Das Konzept der PLANETH-Statistik	157
7.2	Die vom PLANETH-Systemprogramm gelieferten Daten	159
7.2.1	Die Anfangsstanzen	161
7.2.2	Die Auswahlantwortstanzung	161
7.2.3	Die Statistikstanzen bei numerischen Antworten	162
7.2.4	Die Schlussstanzung	163
7.2.5	Die Prüfsummenstanzung am Schluss eines Datensatzes	164
7.3	Das Statistikconverterprogramm STACO	164
7.4	Das PLANETH-Statistikauswertprogramm PLAST	166
7.4.1	Kurze Adressatenstatistik	170
7.4.2	Adressatenstatistik mit automatischer Notengebung	171
7.4.3	Ausführliche Adressatenstatistik	173
7.4.4	Kurze Lehrprogrammstatistik	174
7.4.5	Ausführliche Lehrprogrammstatistik	176

8. BETRIEBSERFAHRUNGEN MIT DEM LEHRSYSTEM PLANETH	179
8.1 PLANETH im Urteil der Studenten	179
8.2 Die Zuverlässigkeit des Lehrsystems PLANETH	182
8.3 Die Kosten beim Lehrsystem PLANETH	184
8.3.1 Die Kosten beim realisierten 6-Konsolen-System	184
8.3.1.1 Die Kosten des gegenwärtigen PLANETH-Betriebs	185
8.3.1.2 Die Betriebskosten des 6-Konsolen-Systems bei Vollauslastung	187
8.3.2 Die Kosten eines voll ausgelasteten 32-Konsolen- PLANETH-Systems	187
9. VERGLEICH ZWISCHEN PLANETH UND EINIGEN ANDERN CUU-SYSTEMEN	189
9.1 Vergleich mit andern kleincomputergesteuerten CUU-Systemen	189
9.2 Beispiele einiger grosser CUU-Systeme	193
9.2.1 Das PLATO IV - System	193
9.2.2 Das Lehrsystem TICCIT	195
9.3 Die Kosten bei PLATO IV im Vergleich zu den Kosten bei PLANETH	196
10. SCHLUSSBETRACHTUNGEN	200
LITERATURVERZEICHNIS	202
LEBENS LAUF	208

ZUSAMMENFASSUNG

Die wichtigsten Hindernisse beim Einsatz von Systemen für den computerunterstützten Unterricht (CUU) sind die erforderliche hohe Anfangsinvestition und der grosse Aufwand für die Herstellung der Lehrprogramme. Je grösser der gesamte Aufwand ist, desto mehr Benutzer sind für einen wirtschaftlich vertretbaren Lehrbetrieb nötig. Infolge des stark dezentralisierten Bildungswesens in vielen westlichen Ländern ist deshalb die Einführung leistungsfähiger, aber teurer Gross-Lehrsysteme noch sehr schwierig. Will man trotzdem schon heute den computerunterstützten Unterricht in der Praxis einsetzen, so muss man sich meist entweder auf kleinere Lehrsysteme beschränken oder grössere Computersysteme mitbenützen, die vorwiegend anderen Zwecken dienen und deshalb nicht speziell für den computerunterstützten Unterricht geeignet sind.

Die vorliegende Arbeit beschreibt die Entwicklung und eingehende praktische Erprobung eines kleincomputergesteuerten audiovisuellen Lehrsystems für den Einsatz im Mittel- und Hochschulunterricht, das sich besonders für naturwissenschaftliche und technische Fächer eignet. Dieses Lehrsystem erhielt den Namen PLANETH (= Programmierbare LehrAnLage der ETH). Es sollte mehr Möglichkeiten als bisher bekannte kleincomputergesteuerte Lehrsysteme bieten und einige Nachteile dieser Systeme vermeiden.

Das Lehrsystem PLANETH besteht aus dem steuernden Kleincomputer mit einigen Peripheriegeräten und maximal 32 daran angeschlossenen Lehrkonsolen, an denen je ein Adressat unabhängig von den anderen Konsolen ein beliebiges Lehrprogramm bearbeiten kann. Die sechs gegenwärtig vorhandenen Lehrkonsolen präsentieren den Lernenden die Informationen, Fragen und Kommentare über zwei Diaprojektoren, ein Kassettentonbandgerät, eine numerische Anzeigeeinheit und einige Lampen. Sie akzeptieren nicht nur Auswahlantworten, sondern auch numerische Freiwahlantworten. Das System erlaubt lineare und beliebig verzweigte Programme. Neben markoffschen und nichtmarkoffschen Verzweigungen sind auch Zufallsverzweigungen möglich. Alle Adressatenreaktionen werden während des Lehrbetriebs fortlaufend registriert und können später mit dem gleichen Computer statistisch ausgewertet werden.

PLANETH-Lehrprogramme können leicht hergestellt und abgeändert werden. Der Lehrstoff ist auf Dias (24mm•36mm) und fakultativ auf Tonkassetten gespeichert. Die Informationen über Struktur und Ablauf der Lehrprogramme werden mit Hilfe von Lochstreifen in den Kernspeicher des Computers eingelesen. Diese Lochstreifen werden mit einem komfortablen Compiler (Uebersetzungsprogramm) erzeugt. Für die Herstellung der Dias genügen eine Schreibmaschine, einige Filzschreiber und eine Kleinbildkamera. Auf den Tonkassetten werden die einzelnen Sprach- und Tonsequenzen mit einem einfachen Gerät markiert.

Ein wesentlicher Vorteil von PLANETH ist die Möglichkeit einer eingehenden statistischen Analyse aller Lehrprogrammdurchläufe. Sie liefert den Lehrprogrammautoren eine ausgezeichnete Rückmeldung über die Qualität der Lehrprogramme und den Lernerfolg der Adressaten. Auf Grund der Ergebnisse dieser statistischen Auswertungen können die einzelnen Lehrprogramme laufend verbessert werden.

Das Lehrsystem PLANETH wird seit September 1974 regelmässig im Hochschulunterricht an der ETH eingesetzt. Es ist bei den Studenten sehr beliebt und wird rege benützt. In den letzten 3½ Jahren wurden auf PLANETH über 5300 Programmdurchläufe absolviert.

PLANETH - A MINICOMPUTER-CONTROLLED INTERACTIVE AUDIOVISUAL SYSTEM

FOR COMPUTER-ASSISTED INSTRUCTION , ADAPTED IN PARTICULAR TO

SCIENTIFIC AND TECHNICAL SUBJECTS AT COLLEGE AND UNIVERSITY LEVEL

ABSTRACT

The most important difficulties when using systems for Computer-Assisted Instruction (CAI) are large initial investments and high costs for producing teachware. The higher the total costs, the more users are needed for an economically viable use of the CAI-system. Due to the decentralized organisation of education in many countries, the introduction of powerful, but expensive large-scale CAI-systems is very difficult. For present day use of CAI, in most cases a limitation to smaller systems is necessary, or larger systems, which are primarily designed for other purposes and not especially suitable for CAI, must be used on a time-sharing basis.

This thesis describes the development and practical evaluation of PLANETH, a minicomputer-controlled audiovisual CAI-system especially suitable for instruction of scientific and technical subjects at college and university level. It offers more possibilities than earlier minicomputer-controlled general-purpose CAI-systems and avoids some disadvantages and limitations of those systems.

PLANETH consists of a minicomputer controlling the whole CAI-system, some peripheral devices and up to 32 student terminals. At each terminal a student can work on a program completely independent from all other users. At present 6 student terminals are available. Informations, questions and comments are presented by 2 random-access slide-projectors, a random-access cassette recorder, a numerical display unit for 8 digits and by some lamps. The CAI-system not only processes multiple-choice answers but also numerical responses. Linear as well as branched programs can be realized. Intrinsic, extrinsic and random branching is possible. While the CAI-system is in use, all student responses are continuously punched on tape for later statistical evaluation by means of the same minicomputer.

PLANETH-teachware can easily be produced and modified. Lesson material is stored on slides (24mm·36mm) and, if sound is used, also on cassettes. The informations about the structure of the teaching programs are read from punched tapes into the computer memory. A very comfortable compiler is available for the generation of these tapes. In most cases only a typewriter, some colored pens and a camera are needed for the production of the slides. A simple apparatus is used to mark each audio-sequence on the audio-tapes.

A very important advantage of PLANETH is the possibility of a thorough statistical analysis of all teaching-program runs. It provides the program authors with excellent feedback about the quality of their teaching-programs and about student performance. Basing on the results of these evaluations, the teaching-programs can be improved continuously.

The PLANETH-system has been in regular use at the Swiss Federal Institute of Technology since September 1974. Students appreciate it very much. More than 5300 program runs were registered during the last 3½ years.

1. VON DER PROGRAMMIERTEN INSTRUKTION ZUM COMPUTER- UNTERSTUETZTEN UNTERRICHT

Seit Jahrhunderten wächst das Wissen der gesamten Menschheit exponentiell und verdoppelt sich etwa alle 15 Jahre. Gleichzeitig veraltet das einmal erworbene Wissen immer rascher. Nach Frank [23] kann man heute mit folgenden durchschnittlichen Nutzungszeiten rechnen:

Berufsbezogenes Fachwissen: 7 Jahre

Hochschullehrstoffe : 14 Jahre (nach andern Autoren 10 Jahre)

Elementarschulstoff : 28 Jahre

Immer mehr Menschen müssen somit immer rascher immer mehr lernen. Während in Wirtschaft und Technik sehr viele Vorgänge rationalisiert und automatisiert wurden, haben sich im Unterrichtswesen bis heute die traditionellen Strukturen weitgehend erhalten.

Erst in den letzten zwei Jahrzehnten wurden grössere Anstrengungen zur Rationalisierung des Unterrichts unternommen. Diese kann vor allem durch eine Objektivierung des Unterrichts erreicht werden, d.h. durch eine Delegation der routinemässigen Arbeit des Lehrers an nicht-menschliche Systeme.

Eine solche Methode zur Unterrichtsobjektivierung ist die programmierte Instruktion (siehe Kap. 1.1). Programmierte Lehrgänge wurden zunächst hauptsächlich in Buchform verwendet. Bald erkannte man, dass die maschinelle Darbietung eines Lehrprogramms auf einem Lehrautomaten gewisse pädagogische Vorteile bietet (siehe Kap. 1.2). Eine noch wesentlich wirksamere Steuerung des Lehrprozesses konnte durch den Einsatz eines Computers erreicht werden. So entstand der computerunterstützte Unterricht (CUU, engl. CAI = Computer Assisted Instruction). Bei Verwendung von leistungsfähigen Grosscomputern ist der CUU heute jedoch nicht mehr auf die Strategie der programmierten Instruktion beschränkt, sondern erlaubt auch andere Lehrstrategien, wie beispielsweise das Lösen von Problemen, Fragelernen, Simulationen und Spiele.

Da der CUU zumindest anfänglich sehr stark von der programmierten Instruktion beeinflusst wurde, soll diese im folgenden Kapitel kurz vorgestellt werden.

1.1 Die programmierte Instruktion

Das Aufkommen der programmierten Instruktion wurde durch die grundlegenden Arbeiten von Skinner [19] und Crowder [21,22] ausgelöst. Bei der programmierten Instruktion wird der Lehrstoff dem Lernenden (=Adressaten) nicht durch einen Lehrer vermittelt, sondern in Form eines Lehrprogramms durch ein mit technischen Mitteln erzeugtes Medium. Dieses Medium kann ein Lehrprogrammbuch, ein Lehrautomat (siehe Kap. 1.2) oder ein Computerterminal sein.

Der Begriff "Programm" wird auch zur Bezeichnung der Steuerinformation für einen Computer benützt. Statt von einem Lehrprogramm spricht man deshalb oft auch von einem Lehralgorithmus, wenn eine klare Unterscheidung nötig ist (vergl. [13]).

Ein Lehrprogramm besteht aus einer Folge von Lehrschritten, bei denen vom Adressaten je eine Reaktion erwartet wird. Ein Lehrschritt besteht zunächst meist aus einer kleinen Portion des zu vermittelnden Lehrstoffs, dem sogenannten Lehrquant, einer Frage und einem Aufruf zu einer Reaktion. Nach einer gewissen Zeit erfolgt die Antwort des Adressaten. Darauf folgt meist sofort eine Beurteilung der Antwort, wobei oft noch die richtige Lösung samt Lösungsweg angegeben wird. Das Arbeitstempo wird dabei durch den Adressaten selbst bestimmt, die programmierte Instruktion ist also zeitadaptiv.

In einem Lehrschritt müssen nicht unbedingt alle oben erwähnten Elemente vorkommen. In einem Übungsprogramm können beispielsweise alle Lehrquanten fehlen und in einem reinen Prüfungsprogramm sogar Lehrquanten, Beurteilungen und Lösungen. Wenn dagegen sehr viel Lehrstoff mit nur wenigen Fragen dazwischen dargeboten werden soll, können auch mehrere Informationslehrschritte ohne Fragen, Beurteilungen und Lösungen unmittelbar aufeinander folgen.

Ueber die zweckmässige Länge und den richtigen Schwierigkeitsgrad eines Lehrschritts sowie über den günstigsten Lehrweg, d.h. die Folge der einzelnen Lehrschritte, gibt es verschiedene Meinungen.

Skinner propagiert beispielsweise eine streng lineare Abfolge der einzelnen Lehrschritte (siehe Fig. 1.1). Diese sollen sehr kurz sein, d.h. nur etwa 20 bis 40 Sekunden dauern. Die zugehörigen Fragen, die meist aus einer auszufüllenden Lücke im Lehrtext bestehen, sollen von 90 - 95 % aller Adressaten richtig beantwortet werden können (Postulat der Kleinstlehrschritte, vergl. [13]). Da der Lernende sehr oft richtig

antwortet, erhält er dank der sofortigen Beurteilung sehr viele sogenannte positive Verstärkungen, was die Lernmotivation und den Lernerfolg günstig beeinflusst [19]. Da alle Adressaten alle Lehrschritte durcharbeiten müssen, sind Lernzeitunterschiede nur auf Grund unterschiedlicher Bearbeitungszeiten der einzelnen Lehrschritte möglich.

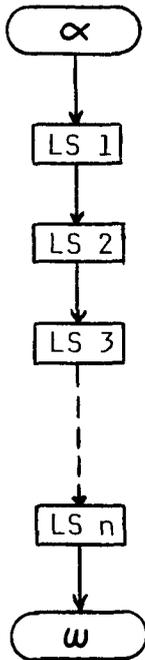
Crowder hingegen vertritt die Ansicht, dass die Antworten des Adressaten auf (nicht allzu einfache) Fragen primär zur Steuerung des Lehrprogrammablaufs verwendet werden sollen, da sie Aufschluss darüber geben, ob der Lehrprozess erfolgreich verläuft [22]. Das Lehrprogramm kann so auf Grund der Reaktionen des einzelnen Adressaten automatisch modifiziert werden, bis dieser das gewünschte Lehrziel erreicht hat. Auf diese Weise wird der Tatsache besser Rechnung getragen, dass das Lernen ein sehr individueller Vorgang ist, der zahlreichen Störeinflüssen unterliegt und der auch viele Ähnlichkeiten mit einem Regelkreis hat (siehe [6, 13, 14, 26]).

Verzweigte Crowder-Lehrprogramme sind somit nicht nur zeitadaptiv, sondern auch topologisch adaptiv, d.h. der Lehrweg ist für verschieden reagierende Adressaten unterschiedlich (Postulat der Wegadaptivität, vgl. [13]). Aus praktischen Gründen werden bei verzweigten Lehrprogrammen besonders bei einfachen Darbietungsmedien (Lehrprogrammbuch, Lehrautomaten) häufig Fragen mit Auswahlantworten oder kurz Auswahlfragen verwendet, da Auswahlantworten mit geringem Aufwand voneinander unterschieden werden können. Dabei besteht jedoch eine gewisse Schwierigkeit darin, gute Falschantworten zu finden, die für gewisse typische Fehler charakteristisch sind, so dass der Adressat vor echte Entscheidungen gestellt wird.

Die Struktur eines Lehrprogramms lässt sich in Form eines Flussdiagramms darstellen, das die Folge der Lehrschritte in Funktion der Adressatenreaktionen angibt. Fig. 1.1 zeigt je ein Flussdiagramm und die charakteristischen Merkmale eines linearen Skinner-Lehrprogramms und eines verzweigten Crowder-Lehrprogramms. Selbstverständlich können neben diesen reinen Formen auch beliebige Mischformen realisiert werden.

Beispielsweise kann man bei einem linearen Lehrprogramm echte Fragen mit Auswahlantworten verwenden, nach einer falschen Antwort den Adressaten einen oder mehrere Lehrschritte wiederholen lassen oder in gewissen Fällen Umwege vorsehen. Bei einem verzweigten Lehrprogramm kann man auch sogenannte konstruierte oder Freiwahlantworten vorsehen, sofern das verwendete Darbietungsmedium solche Antworten verarbeiten kann. Wenn man vom Exaktvergleich absieht, der auch auf gewissen Lehrautomaten in beschränkter Form möglich ist, erfordert die automatische Analyse von Freiwahlantworten

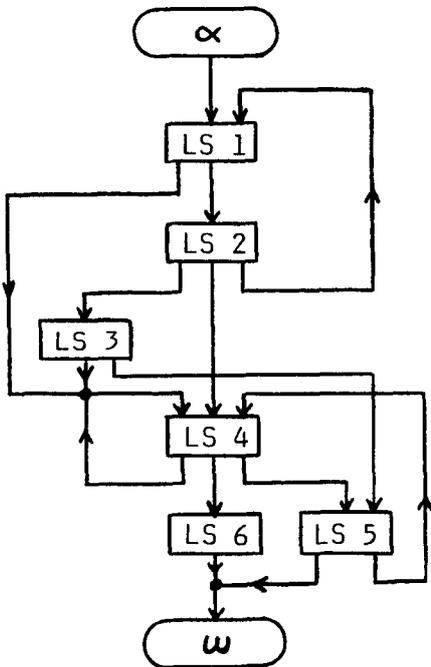
Lineares Lehrprogramm:



Merkmale (nach Skinner):

- Kleinstlehrschrirte
- Sehr leichte Fragen, damit 90-95% richtige Antworten auftreten
- Freiwahlantworten (Lücke im Lehrtext ausfüllen)
- Sofortige Antwortbeurteilung und somit häufige Verstärkungen
- Lehrprogramm zeitadaptiv

Verzweigtes Lehrprogramm:



Merkmale (nach Crowder):

- Lehrschrittgrösse variabel
- Echte Fragen, Fehlerhäufigkeit variabel
- Auswahlantworten
- Sofortige Antwortbeurteilung, Verstärkungen bei richtigen Antworten
- Lehrprogramme zeitadaptiv und topologisch adaptiv

Legende : LS = Lehrschritt

α = Lehrprogramm anfang

ω = Lehrprogrammende

Fig. 1.1 : Flussdiagramm eines linearen und eines verzweigten Lehrprogramms

jedoch immer den Einsatz eines Computers. Schliesslich können in einem Lehrprogramm auch lineare und verzweigte Programmteilstücke kombiniert werden.

Die Struktur eines Lehrprogramms kann jedoch nur bei sogenannten markoffschen [13] oder intrinsischen [22] Lehralgorithmen durch ein Flussdiagramm anschaulich dargestellt werden. Bei einem markoffschen resp. intrinsischen Lehralgorithmus ist der nächste Lehrschritt jeweils nur durch den vorangegangenen und die Reaktion des Adressaten auf diesen bestimmt. Bei einem nichtmarkoffschen resp. extrinsischen Lehralgorithmus ist der jeweils folgende Lehrschritt ausser vom letzten Lehrschritt und der darauf folgenden Reaktion auch vom weiter zurückliegenden Adressatenverhalten und somit von der Lerngeschichte abhängig. Nichtmarkoffsche Lehrprogramme können sich den individuellen Unterschieden zwischen den einzelnen Adressaten deshalb noch besser anpassen. Ausser in ganz einfachen Fällen können nichtmarkoffsche Lehrprogramme aber nur auf computergesteuerten Lehrsystemen realisiert werden.

Der Einsatz der programmierten Instruktion bietet folgende Vorteile (vergl. [23, 24, 25]).

1. Höhere Lernwirksamkeit. Nach Frank ist der Wirkungsgrad der programmierten Instruktion gegenüber dem konventionellen Klassenunterricht um 10-20%, gegenüber der traditionellen Hochschulvorlesung sogar bis 50% grösser. Dies dürfte auf die wesentlich stärkere Eigenaktivität der Lernenden, die Anpassung an ihren individuellen Arbeitsrhythmus sowie die gegenüber dem konventionellen Unterricht sehr viel bessere Vorbereitung der Lehrprogramme zurückzuführen sein.
2. Die Lernintimität. Besonders bei schwierigen oder von starken Tabus umgebenen Lehrstoffen erweist sich das sachliche Verhältnis zu einem vom Lehrer unabhängigen Lehrsystem oft als günstiger, da der Adressat nicht befürchten muss, sich bei Lernschwierigkeiten vor dem Lehrer oder vor Mitschülern blosszustellen.
3. Die Zuverlässigkeit. Die Lernwirksamkeit eines gründlich getesteten Lehrprogramms ist mit viel grösserer Wahrscheinlichkeit voraussagbar als diejenige der Direktschulung.
4. Die Wirtschaftlichkeit. Wenn genügend grosse Adressatenzahlen vorhanden sind, ist der Aufwand für das Erstellen einer Lehrprogrammlektion kleiner als für das wiederholte Unterrichten desselben Lehrstoffes. In der einschlägigen Fachliteratur schwanken die Angaben

über den notwendigen Zeitaufwand für das direkte Herstellen eines Lehrprogramms allerdings sehr stark (Minimum 40 Std., Maximum 3900 Std. pro Lehrprogrammstunde). Frank [23] gibt für audiovisuell gestaltete Lehrprogramme etwa 250 Arbeitsstunden pro Lehrprogrammstunde an. Details über die Wirtschaftlichkeit siehe Kap. 1.4 .

5. Die leichte Nachholmöglichkeit für Adressaten, die den normalen Unterrichtstermin z.B. infolge Krankheit versäumt haben.
6. Die Kenntnisangleichung der Adressaten, die z.B. für eine anschließende weiterführende Direktschulung erwünscht ist.
7. Einsparung von qualifiziertem Lehrpersonal. Der Lehrermangel war Ende der sechziger / Anfang der siebziger Jahre ein ernsthaftes Problem. Infolge des Geburtenrückgangs ("Pillenknick") hat sich die Lage inzwischen auf der Primar-, Sekundar- und Mittelschulstufe weitgehend entspannt. Auf Hochschulstufe ist das Problem jedoch weiterhin vorhanden (drohender Numerus Clausus). Hier können durch den Einsatz objektivierter Lehrverfahren die vorhandenen Professoren und Assistenten vor allem bei der Vermittlung von grundlegenden Kenntnissen entlastet werden.

Die oben erwähnten Vorteile sind immer vorhanden, gleichgültig ob als Darbietungsmedium ein Lehrprogrammbuch, ein Lehrautomat oder ein computergesteuertes Lehrsystem verwendet wird. Natürlich bringt der Einsatz von Lehrautomaten und Computern weitere wesentliche Vorteile, die in den folgenden Kapiteln kurz behandelt werden.

1.2 Lehrautomaten

Programmierte Instruktion wird auch heute noch zu einem grossen Teil mit Hilfe von Lehrprogrammbüchern durchgeführt. Sie sind billig und erfordern keine speziellen technischen Apparaturen, also keine hohen Anfangsinvestitionen. Lehrprogrammbücher haben aber noch wesentliche lehrtechnische Mängel. Durch den Einsatz von Lehrautomaten können viele davon vermieden werden.

Unter einem Lehrautomaten versteht man ein Gerät, das mindestens folgende **Lehrfunktionen** automatisch durchführen kann :

- a) Speicherung und Darbietung des Lehrstoffs.
- b) Analyse der Adressatenreaktion (Richtigkeitsprüfung)
- c) Programmgemässes Aufsuchen des nächsten Lehrschritts auf Grund der Adressatenreaktion.

Der Einsatz von Lehrautomaten bietet gegenüber Lehrprogrammbüchern folgende **Vorteile** (vergl. [5, 23]).

1. Die Möglichkeit, auch akustische Information zu übertragen.
2. Die Möglichkeit, Laufbilder darzubieten.
3. Die Sicherheit vor Mogelversuchen, d.h. es ist dem Adressaten nicht möglich, die richtige Lösung ohne eigenen Lösungsversuch zu finden.
4. Die Möglichkeit, die Folge der Adressatenreaktionen und den Lehrweg zu registrieren (z.B. für Testzwecke oder zur Lehrprozesssteuerung).
5. Durch die automatische Richtigkeitsprüfung und das automatische Aufsuchen des nächsten Lehrschritts können wertvolle Lernzeit eingespart und viele Fehlermöglichkeiten eliminiert werden. Gleichzeitig wird dadurch der sogenannte "Brockhaus-Effekt" (zufälliges Hängenbleiben an auffälligen Bild- oder Textstellen beim Aufschlagen einer neuen Seite im LehrprogrammBuch [13]) mit Sicherheit vermieden.

Bei praktisch realisierten Lehrautomaten sind meist nicht alle der fünf angegebenen Vorteile wirklich vorhanden. So verzichten viele Lehrautomaten wegen des relativ hohen Aufwandes auf bewegte Bilder (Punkt 2), da solche meist nur selten gebraucht werden. Punkt 4 (Lehrwegregistrierung) lässt sich mit Lehrautomaten wegen des dafür benötigten hohen Speicherplatzbedarfs nur schwer und sehr rudimentär realisieren, besonders wenn die gespeicherten Daten auch noch zur Lehrprozesssteuerung verwendet werden sollen. Viele Lehrautomaten verzichten deshalb von vorneherein auf diese Möglichkeit.

Der erste Lehrautomat für die Durchführung linearer Lehrprogramme mit 4 Auswahlantworten wurde von Pressey bereits 1926 konstruiert [37], seine Arbeiten gerieten aber wieder in Vergessenheit. Einige Beispiele für modernere Lehrautomaten sind u.a. der amerikanische "Auto Tutor Mark II", der in [22] kurz beschrieben ist, das tschechische System "Unitutor" [27], das deutsche Lehrgerät BASF LG 5100 [14] und die erste mikroprozessorgesteuerte Lehrmaschine "STA TM 150" aus schweizerischer Produktion [16].

Bei autonomen (d.h. computerunabhängigen) Lehrautomaten lässt sich die Methode der Auswahlantworten technisch relativ einfach realisieren. Jeder Auswahlantwort ist eine bestimmte Antworttaste zugeordnet. Der Adressat teilt der Lehrmaschine durch das Drücken der entsprechenden Taste mit, welche Antwort ihm richtig erscheint. Die Steuerelektronik veranlasst darauf den Transport des oder der Lehrstoffspeicher (Diaprojektor oder Filmstreifenprojektor, Tonbandgerät usw.) zum Anfang des nächsten Lehrschritts. Je nach der eingegebenen Antwort wird eine bestimmte Sprungweite und Sprungrichtung ausgelöst. Auf diese Weise können mit geringem Aufwand verzweigte markoffsche Lehrprogramme dargeboten werden, denn für die Kennzeichnung der richtigen Antworten muss nur sehr wenig Information gespeichert werden.

Im Gegensatz zu Auswahlantworten sind Freiwahlantworten mit autonomen Lehrautomaten nur schwer zu verarbeiten, da zunächst mehrere Tastendrucke gespeichert werden müssen, bevor die Richtigkeitsprüfung erfolgen kann. Dies erfordert einen wesentlich höheren Aufwand bei der Steuerelektronik, der infolge der fehlenden Rechenmöglichkeit dann aber trotzdem höchstens einen Exaktvergleich (lückenlose zeichenweise Übereinstimmung zwischen Adressatenantwort und richtiger Lösung) erlaubt. Durch den Einsatz von Mikroprozessoren könnte dieser Mangel in den kommenden Jahren jedoch zumindest teilweise behoben werden (vergl. [16]).

1.3 Der computerunterstützte Unterricht (CUU)

Die aufwendigste Form objektivierter Schulung ist zweifellos der computerunterstützte Unterricht. Ein computergesteuertes Lehrsystem besteht aus einem (gelegentlich mehr als einem) Computer, der die ganze Anlage zentral steuert, und meist mehreren mit diesem verbundenen Adressatenarbeitsplätzen (= Terminals, = Konsolen).

Der Computer ist in der Lage, alle während des Lehrprozesses anfallenden didaktisch relevanten Daten zu speichern und auf Grund einer laufenden Analyse dieser Daten den Lehrprogrammablauf für jeden einzelnen Adressaten adaptiv zu steuern. Es lassen sich somit leicht nichtmarkoffsche resp. extrinsische Lehralgorithmen realisieren, die wesentlich besser auf die individuellen Unterschiede im Lernverhalten eingehen können (siehe Kap. 1.1).

Die laufende Registrierung dieser Lehrprozessdaten erlaubt zudem eine eingehende statistische Analyse der Lehrprogrammdurchläufe. Dies ermöglicht beispielsweise eine laufende Ueberwachung der Lernwirksamkeit der einzelnen Lehrprogramme und sogar der einzelnen Lehrschritte auf Grund von sehr vielen Durchläufen. Wenn nötig, können die einzelnen Lehrprogramme darauf leicht modifiziert werden. Bei Lehrprogrammbüchern und autonomen Lehrautomaten ist die Validierung von stark verzweigten Lehrprogrammen viel problematischer, denn es muss in der Phase der Gruppenerprobung mit relativ wenigen Testpersonen gearbeitet werden. Ist das Buch- resp. Automatenprogramm einmal erstellt, sind Korrekturen und Aenderungen nur mit sehr hohem Aufwand durchführbar (siehe z.B. [14]).

Der Computer kann aber nicht nur für die Steuerung des Lehrprozesses und die Auswertung der anfallenden Statistikdaten eingesetzt werden, sondern leistet auch wertvolle Hilfe bei der Produktion der Lehrprogramme. Besonders der Aufwand für die Herstellung und das Austesten des Lehrprogrammcodes, der die für die Steuerung des Lehrprogrammablaufs im Computer nötigen Daten enthält, kann durch die Verwendung eines Computers stark reduziert werden (siehe Kap. 6).

Je nach der Grösse des zentralen Computers kann man die computergesteuerten Lehrsysteme in zwei Hauptgruppen einteilen (vergl. [28]). Die erste Gruppe bilden die kleincomputergesteuerten audiovisuellen Lehrsysteme, die zweite Gruppe besteht aus den grosscomputergesteuerten Lehrsystemen, bei denen man weiter eine Untergruppe mit Standard-Terminals und eine Untergruppe mit speziell für CUU geeigneten Spezialterminals unterscheiden kann.

1.3.1 Kleincomputergesteuerte audiovisuelle Lehrsysteme

Diese Lehrsysteme werden von einem Kleincomputer (Speichergrösse $\leq 16K$) ohne zusätzliche Massenspeicher (Plattenspeicher usw.) gesteuert. Dabei ist der Lehrstoff, der sehr viel Computerspeicherplatz belegen würde, in dezentralen, beliebig fernsteuerbaren Bild- und Tonspeichern (z.B. Diaprojektoren und Tonbandgeräten) untergebracht. Jeder Adressatenplatz verfügt über einen solchen Bild- und Tonspeicher. Der Lehrprogrammcode, der die Informationen über Struktur und Ablauf des Lehrprogramms enthält, benötigt dagegen wesentlich weniger Platz und kann deshalb gut im Speicher eines Kleinrechners untergebracht werden. Die Lehrstoffdarbietung erfolgt audiovisuell mit farbigen Bildern und Ton. Für die Eingabe der Adressatenreaktionen stehen

an jedem Arbeitsplatz einige Antworttasten zur Verfügung. Dank der Möglichkeit, farbige Fotografien und komplexe Zeichnungen und Grafiken einzusetzen, die dank der Verwendung mehrerer Farben noch übersichtlich sind, lassen sich oft komplizierte Sachverhalte anschaulich darstellen, die mit Worten allein nicht richtig beschrieben werden können.

Solche Lehrsysteme sind bestens geeignet für die Steuerung auch sehr komplexer Lehrprogramme mit Auswahlantworten. Die Rechenkapazität und Speichergrosse des Kleincomputers erlaubt auch die Verarbeitung von sogenannten konstruierten oder Freiwahlantworten, wenn diese nicht in allzu komplizierter Form vorliegen (z.B. als Zahl) oder wenn relativ einfache Verfahren (z.B. Exaktvergleich einzelner Wörter) für die Auswertung der Antwort ausreichen.

Beispiele solcher kleincomputergesteuerter audiovisueller Lehrsysteme, die jedoch nur Auswahlantworten verarbeiten können, sind das System EDUCATOR [15] und das Lehrsystem der SWISSAIR [30].

1.3.2 Grosscomputergesteuerte Lehrsysteme

Diese Lehrsysteme werden von einem (evtl. mehr als einem) mittleren oder grossen Computer mit einem oder mehreren zusätzlichen Massenspeichern (z.B. Plattenspeichern) gesteuert. Der finanzielle Aufwand für den Computer und seine Peripherie ist deshalb sehr viel (z.B. mehrere hundert mal) grösser als bei kleinrechnergesteuerten Systemen. Der Lehrstoff ist zusammen mit dem Lehrprogrammcode meist auf den Massenspeichern untergebracht.

Je nach den verwendeten Terminals kann man zwei Gruppen von grosscomputergesteuerten Lehrsystemen unterscheiden.

Die erste Gruppe verwendet Standard-Terminals, wie sie auch in der übrigen Datentechnik benützt werden. Meist handelt es sich um alphanumerische Bildschirmterminals (eventuell mit zusätzlichen grafischen Ein- und Ausgabemöglichkeiten) oder um Fernschreiber. Solche Terminals können nur schwarz-weiße Darstellungen liefern. Die Eingabe der Adressatenreaktionen erfolgt über eine alphanumerische Tastatur und eventuell mit Hilfe eines Lichtstifts.

Bei Verwendung von Standard-Terminals ist somit eine gewisse Verarmung der Darbietungsmöglichkeiten gegenüber andern Medien festzustellen (vergl. [28]), besonders wenn keine besonderen grafischen Darstellungsmöglichkeiten vorhanden sind. Solche Terminals eignen sich gut für die Vermittlung stark formalisierter Lehrstoffe wie Mathematik und Kurse über Computerprogrammierung. Für andere Lehrstoffe, die auch farbige Bilder, nicht durch einfache mathematische Funktionen beschreibbare grafische Darstellungen und Ton benötigen, sind Lehrsysteme mit Standard-Terminals weniger gut geeignet. Solche Fachgebiete sind z.B. Medizin, Natur- und Ingenieurwissenschaften.

Grosscomputergesteuerte Lehrsysteme mit Standard-Terminals wurden deshalb vorwiegend von Computerherstellern und von Informatikinstituten oder Rechenzentren an Hochschulen entwickelt. Bei diesen Informatikinstituten steht in der Lehre natürlich die Vermittlung von Programmierkenntnissen im Vordergrund, und sie haben für ihre übrige Arbeit ohnehin einen Grosscomputer mit vielen Standard-Terminals zur Verfügung. Solche Lehrsysteme werden oft so konzipiert, dass sie neben dem normalen Rechenbetrieb auf dem Grosscomputer laufen können. Dies hat aber den Nachteil, dass Systemzusammenbrüche, die bei grossen Rechenzentren bekanntlich nicht allzu selten sind, den Lehrbetrieb ebenfalls empfindlich stören. Bei Lehrsystemen, die speziell für den Unterricht im Programmieren geeignet sind, besteht oft auch die Möglichkeit, vom Lernenden am Terminal erzeugte Programmstücke vom Computer während des Lehrbetriebs ausführen zu lassen und die Resultate gleich wieder auf das Terminal auszugeben. Beispiele solcher an Informatikinstituten entwickelter Lehrsysteme sind LEKTOR [31], LEGIS [8] und THALES [9].

Die zweite Gruppe von grosscomputergesteuerten Lehrsystemen verwendet modifizierte Standardterminals oder speziell für CUU entwickelte Spezialterminals. Ein solches modifiziertes Terminal kann beispielsweise aus einer normalen Bildschirmkonsole bestehen, die zusätzlich über einen Diaprojektor und ein Tonbandgerät verfügt, die vom Computer aus beliebig fernsteuerbar sind. Speziell für CUU konzipierte Terminals besitzen oft einen Farbfernsehschirm für die Ausgabe und eine alphanumerische Tastatur für die Eingabe (z.B. bei TICCIT [32] und GENIUS [33]). Eine besonders raffinierte Lösung ist die Plasma-Display-Konsole des PLATO IV - Systems [34], welche die Uebereinanderprojektion von farbigen Bildern (auf Mikrofilm gespeichert) mit computergenerierten Texten und Grafiken gestattet, ohne dass in der Konsole spezielle Datenspeicher nötig sind. Die Eingabe der Adressatenantwort erfolgt bei der PLATO IV - Konsole in der Regel

über eine alphanumerische Tastatur, kann jedoch auch durch Berühren einzelner Stellen auf dem Plasma-Bildschirm geschehen.

Solche Lehrsysteme mit Spezialterminals eignen sich für sehr viele Fachgebiete und Lehrstrategien. Sie präsentieren den Lehrstoff vorwiegend visuell, bei einigen Systemen (TICCIT, PLATO IV u.a.) ist jedoch in beschränktem Umfang zusätzlich auch die Darbietung von akustischer Information möglich. Meist werden Standbilder verwendet, denn für bewegte Bilder wird nochmals wesentlich mehr Speicherplatz benötigt. Computergenerierte Texte und Grafiken können mit farbigen Bildern kombiniert werden, die teils zentral in digitaler Form auf Massenspeichern (TICCIT), teils dezentral auf Diapositiven oder Mikrofilmkarten (PLATO IV) in den einzelnen Terminals gespeichert sind.

Diese dezentrale Form der Bildspeicherung entlastet natürlich den Computer, den zentralen Massenspeicher und die Ein-/Ausgabekanäle wesentlich, denn für die digitale Speicherung komplizierter farbiger Bilder wird sehr viel Platz benötigt.

1.3.3 Neuere Lehrstrategien bei grossrechnergesteuerten Lehrsystemen

In den sechziger Jahren wurde zunächst auch mit grosscomputergesteuerten Lehrsystemen sehr viel programmierte Instruktion durchgeführt. Manche Computerhersteller entwickelten für diese Anwendung sogar spezielle CUU-Sprachen, sogenannte Autorensprachen (z.B. COURSEWRITER von IBM, COPI von UNIVAC).

Seit Beginn der siebziger Jahre hat man jedoch erkannt, dass Grosscomputer auf diese Weise nur schlecht ausgenützt werden. Moderne grosscomputergesteuerte Lehrsysteme erlauben heute dank der hohen Rechengeschwindigkeit und riesigen Speicherkapazität eine grosse Vielfalt verschiedener Lehrstrategien.

Neben programmierter Instruktion, Übungs- und Unterweisungsprogrammen sind folgende Lehrformen möglich :

- Problemlösen : Der Lernende bearbeitet unter der Anleitung des Lehrsystems relativ komplexe Probleme, die individuell für ihn erzeugt wurden und lässt seine Lösung anschliessend vom Computer beurteilen.

- Simulationen : Der Adressat kann eine Vielzahl von Vorgängen, die er nicht direkt beobachten oder durchführen kann, weil sie zu schnell, zu langsam, zu gefährlich oder zu kostspielig sind, mit Hilfe von ihm gewählter Parameter simulieren. Auf diese Weise gewinnt er ein tiefes Verständnis für den untersuchten Prozess.
Beispiel: Simulierte Experimente über die Vererbungsgesetze in der Biologie bei PLATO IV .
- Spiele : Sie stellen gewissermassen aufgelockerte und weniger zielgerichtete Formen von Simulationen dar, die aber trotzdem sehr lehrreich sein können.
- Fragelernen : Der Lernende versucht, ein unbekanntes Objekt möglichst genau zu bestimmen, indem er dem Lehrsystem Fragen nach bestimmten Eigenschaften des gesuchten Objekts stellt. Der Computer beantwortet diese Fragen so gut wie möglich. Auf diese Weise erhält der Adressat in kurzer Zeit die Resultate sehr vieler simulierter Experimente, denn in der Praxis könnte er die Eigenschaften natürlich nicht erfragen, sondern müsste sie experimentell feststellen. Der Student lernt so, sinnvolle Reihen von Experimenten für die Bestimmung des unbekanntes Objektes anzusetzen.
Beispiele: Bestimmung einer unbekanntes chemischen Substanz oder Erstellung einer Diagnose in der Medizin bei PLATO IV .
- Lernersteuerung: Bei dieser Lehrform wird der Lehrweg statt von Lehrprogramm resp. Programmautor weitgehend durch den Lernenden selbst bestimmt. Er steht in einem regen Dialog mit dem Lehrsystem und kann sich durch gezielte Anfragen an ein darin integriertes Informationssystem den gewünschten Lehrstoff selbst zu einem individuellen Programm zusammenstellen [8, 28, 29, 36].

Diese neueren CUU-Formen sind nicht mehr nur ein- und ausgabeintensiv, sondern oft auch sehr rechenintensiv und benötigen enorm viel Speicherplatz. Sie können deshalb nur auf teuren Grosscomputersystemen realisiert werden.

Der früher oft als Vorteil von Grosscomputerlehrsystemen herausgestrichene Tischrechnermodus, bei dem der Adressat sein Terminal als Tischrechner für numerische Berechnungen benützen konnte, ist heute nicht mehr aktuell. Wegen des Preiszerfalls bei den Taschenrechnern verfügt heute jeder Mittel-

und Hochschüler über einen eigenen Taschenrechner, dessen Bedienung er dank häufiger Übung viel sicherer beherrscht als den nur relativ selten benützten Tischrechnermodus des Lehrsystems.

Die in den letzten Jahren aufgekommene Lernersteuerung versucht den Adressaten stärker zu motivieren, indem sie ihm ein entdeckendes Lernen ermöglicht. Damit der Lernende jedoch den Lehrprozess überhaupt sinnvoll beeinflussen kann, muss er erst eine mehr oder weniger komplizierte Schülerkommandosprache erlernen und sich zudem einen gewissen Ueberblick über das entsprechende Fachgebiet verschaffen. Nicht alle Adressaten sind dazu in der Lage, deshalb muss auch in lernergesteuerten Lehrprogrammen eine Autorenssequenz, also ein vom Autor bestimmter Musterlernweg, vorhanden sein, die der Lernende durcharbeiten kann, wenn er die Orientierung verloren hat [8]. Auch Testsequenzen, die der Adressat auf Wunsch verlangen kann, müssen vom Autor vorprogrammiert sein. Nach einem solchen Zwischentest folgen zwar keine zwangsweisen Programmfortsetzungen, sondern es wird dem Lernenden empfohlen, an gewissen sinnvollen Stellen im Programm weiterzufahren. Auch solche Empfehlungen müssen aber vom Autor vorprogrammiert werden. Wenn man dann noch die notwendige komplexe Struktur des den Lehrprozess steuernden Betriebssystems berücksichtigt, erscheint die Behauptung Freibichlers [29] sehr fragwürdig, dass für das Erstellen von lernergesteuerten CUU-Programmen nur ein Bruchteil der Zeit benötigt werde, die für die Herstellung programmgesteuerter Lehrprogramme erforderlich sei. Für Lehrstoffe, die viele komplizierte Bilder und grafische Darstellungen erfordern, dürften im Gegenteil lernergesteuerte Programme wesentlich aufwendiger sein.

Es muss auch darauf hingewiesen werden, dass die Lernersteuerung möglicherweise die Unterschiede zwischen guten und schlechten Adressaten noch vertieft, denn erfolgreich Lernende mit Interesse am Lehrstoff sind zweifellos besser in der Lage, sich einen sinnvollen Lehrweg selbst zusammenzustellen. Zudem können vom Adressaten beschrittene Irrwege die Lernzeit erhöhen und unter Umständen bewirken, dass sich beim Lernenden falsche Vorstellungen festsetzen.

Lernergesteuerte CUU-Programme eignen sich wohl in erster Linie für Mathematikurse und Kurse über Computerprogrammierung, da die entsprechenden Adressaten nicht allzu grosse Schwierigkeiten beim Erlernen der Schülerkommandosprache haben dürften. Auch ist der Lehrstoff durch Text und einfache Grafiken darstellbar.

Bei andern Lehrstoffen mit vielen komplizierten Bildern ist die Programmierung schwieriger, die in Frage kommenden Programmautoren sind mit dem komplexen Betriebssystemprogramm weniger vertraut und die Lernenden haben mehr Mühe, sich die Befehle der Schülerkommandosprache anzueignen. Lehrsysteme, bei denen die Lernersteuerung im Vordergrund steht, wurden deshalb in erster Linie von Informatikinstituten an Hochschulen realisiert [8, 31] .

1.4 Sinnvoller Einsatz des computerunterstützten Unterrichts

Auch die besten heute verfügbaren CUU-Systeme weisen noch wesentliche Beschränkungen auf, die sie nicht für alle Lehrstoffe gleich gut geeignet machen. Es dominiert eindeutig die visuelle Kommunikation zwischen Lehrsystem und Adressat.

Die Ausgabe von Lehrinformation an den Adressaten kann visuell oder akustisch erfolgen. Besonders einfach ist die Vermittlung von Lehrstoffen, die sich durch Buchstaben und Zahlen, also alphanumerische Symbole, darstellen lassen. Die Verwendung spezieller visueller Symbole (z.B. Integralzeichen) bereitet bei manchen Lehrsystemen jedoch schon Schwierigkeiten. Diese Schwierigkeiten werden noch weitaus grösser, wenn neben Text auch grafische Darstellungen und (eventuell sogar bewegte) farbige Bilder präsentiert werden sollen. Immerhin kann man feststellen, dass heute bei genügendem technischem Aufwand für jede Anwendung eine brauchbare visuelle Darbietungsmöglichkeit realisiert werden kann.

Die Ausgabe akustischer Information ist bereits wesentlich problematischer. Da bisher noch keine für Lehrzwecke brauchbaren sprechenden Terminals entwickelt wurden, ist die Ausgabe von akustischer Sprachinformation, welche direkt vom Computer erzeugt wurde, noch nicht möglich. Dagegen können extern aufgenommene Sprach- und Tonsequenzen, die in analoger oder digitaler Form gespeichert sind, von entsprechend ausgerüsteten Lehrsystemen nach Bedarf einzeln adressiert und abgespielt werden.

Für die Eingabe der Adressatenreaktionen in den Computer bestehen noch wesentlich stärkere Einschränkungen. Die meisten CUU-Lehrsysteme verfügen zu diesem Zweck höchstens über eine alphanumerische Tastatur, welche die Eingabe von speziellen visuellen Symbolen nicht gestattet. Wohl verfügen einige Systeme noch über die Möglichkeit der Eingabe grafischer Darstel-

lungen mit dem Lichtstift. Ausser in ganz einfachen Fällen ist jedoch die Auswertung der eingegebenen Figuren sehr kompliziert oder sogar unmöglich. Die Eingabe von akustischen Antworten ist vorläufig noch nicht möglich, da noch kein brauchbares Verfahren zur computergerechten Aufbereitung der menschlichen Sprache existiert.

CUU-Lehrsysteme sollten also nur für Lehrstoffe eingesetzt werden, für die sie auf Grund ihrer technischen Möglichkeiten auch wirklich geeignet sind.

1.4.1 Die Kosten pro CUU-Stunde

Eine zentrale Rolle beim praktischen Einsatz des CUU spielen die meist sehr grossen Anfangsinvestitionen und die immer noch relativ hohen Kosten pro Betriebsstunde. In diesem Kapitel sollen die Kosten k_{CUU} pro "Studentenkontaktstunde" untersucht werden. Diese Kosten setzen sich hauptsächlich aus drei Komponenten zusammen :

- a) Kostenanteil k_A für die Amortisation der technischen Installationen (Hardware) des Lehrsystems,
- b) Kostenanteil k_B für Betriebskosten, d.h. für die Durchführung des Lehrbetriebs inkl. Betreuung der Adressaten sowie Wartung der Anlage,
- c) Kostenanteil k_L für die Entwicklung des entsprechenden Lehrprogramms.

$$\text{Es gilt also: } k_{CUU} = k_A + k_B + k_L = \underbrace{\frac{A}{h_{AK}}}_{k_A} + \underbrace{\frac{B}{h_{AK}}}_{k_B} + \underbrace{\frac{L}{d_{BL}}}_{k_L} \quad (1.1)$$

wobei:

k_{CUU} = Kosten pro CUU-Stunde ("Studentenkontaktstunde")

A = I_H/d_A = Jährliche Kosten für Hardware-Amortisation oder -Miete

I_H = Investitionen für Hardware (technische Installationen)

d_A = Abschreibungsdauer für Hardware in Jahren

h_{AK} = Totale Anzahl Adressatenkontaktstunden pro Jahr

B = Jährliche Lehrsystem-Betriebskosten

L = Kosten für die Lehrprogrammentwicklung

d_{BL} = Totale Benützungsdauer des betreffenden Lehrprogramms in Stunden

Bevor irgendwelche CUU-Versuche durchgeführt werden können, müssen zuerst die meist sehr hohen Hardware-Investitionskosten I_H aufgebracht werden. Damit der Kostenanteil k_A infolge dieser hohen Festkosten nicht allzu gross wird, muss eine möglichst grosse Anzahl Adressatenkontaktstunden (h_{AK}) auf dem Lehrsystem absolviert werden. Die jährlichen Lehrsystem-Betriebskosten B dürften für ein gegebenes Lehrsystem etwa proportional zur Benützungsdauer sein, d.h. k_B kann in erster Näherung als konstant angesehen werden.

Für den Entscheid, für ein schon anderweitig einigermaßen häufig benütztes Lehrsystem ein neues Lehrprogramm zu entwickeln, ist in erster Linie der zu erwartende Kostenanteil k_L massgebend (siehe Formel (1.1)). k_B ist ja ungefähr konstant, und k_A spielt als Festkostenanteil in diesem Fall keine Rolle. k_L ist natürlich gemäss Formel (1.1) umso geringer, je grösser die zu erwartende totale Benützungsdauer d_{BL} des betreffenden Lehrprogramms ist, bevor es wegen Veraltens des Lehrstoffs ersetzt werden muss. k_L darf jedoch nicht absolut betrachtet werden, sondern es ist auch zu berücksichtigen, welche Unterrichtsform auf welcher Stufe durch das CUU-Programm objektiviert werden soll.

1.4.2 Die Rentabilitätsgrenze neuer Lehrprogramme für bereits laufende Lehrsysteme

Für die Beurteilung der Rentabilität neuer Lehrprogramme auf einem bereits laufenden CUU-System muss man die variablen Kostenanteile k_B und k_L gemäss Formel (1.1) mit dem Kostenanteil k_D für eine Stunde Direktunterricht vergleichen. Man erhält dann folgende Ungleichung :

Der Einsatz eines neuen Lehrprogramms auf einem bestehenden CUU-System ist rentabel, wenn

$$k_D \geq k_B + k_L \quad (1.2)$$

wobei : k_D = Kostenanteil für 1 Stunde Direktunterricht

k_B = Betriebskostenanteil für CUU-System

k_L = Kostenanteil für Lehrprogrammentwicklung

Für Lehrsysteme mit einem vernünftig kleinen k_B -Wert lässt sich die Rentabilitätsgrenze leicht annähernd abschätzen :

Die Erstellung eines neuen Lehrprogramms ist dann rentabel, wenn die für die Lehrprogrammherstellung benötigte Arbeitszeit T_L des Lehrers (inkl. evtl. Hilfskräfte) kleiner ist als die Arbeitszeit für die direkte Schulung einer

entsprechenden Anzahl Adressaten.

Die Rentabilitätsgrenze für den Einsatz eines neuen Lehrprogramms auf einem bereits benützten Lehrsystem ist somit näherungsweise durch folgende Ungleichung definiert (vergl. Frank [13]) :

$$n_K \cdot T_D \geq T_L \quad \text{oder} \quad \frac{a}{g} T_D \geq T_L \quad (1.3)$$

wobei : $n_K = a/g =$ Anzahl direkt zu schulende Klassen
 $a =$ Anzahl zu schulende Adressaten
 $g =$ Mittlere Schulungsgruppenstärke (Klassengrösse)
 $T_D =$ Arbeitszeit des Lehrers (inkl. evtl. Hilfskräfte) für Vorbereitung und Durchführung einer Stunde Direktunterricht
 $T_L =$ Arbeitszeit des Lehrers (inkl. evtl. Hilfskräfte) für die Herstellung eines einstündigen Lehrprogramms

Alle diese Grössen sind natürlich stark vom Lehrstoff und von der Schulstufe abhängig. Frank [13] gibt für T_D 2 Stunden an. Dieser Wert dürfte für die Primar- und Sekundarstufe stimmen. Für eine Hochschulvorlesung, bei der meist auch 1 - 2 Assistenten mitwirken und die periodisch überarbeitet werden muss, ist für T_D ein Wert von 4 - 6 Stunden jedoch realistischer. Für Hochschulübungen in Gruppen, die von Assistenten durchgeführt werden, dürfte T_D etwa 3 Stunden betragen.

Die Angaben für T_L in der Literatur schwanken ausserordentlich stark (siehe Kap. 1.1). Nach unseren Erfahrungen (siehe Kap. 6.4) sind für Programme mittlerer Komplexität Werte von 100 - 150 Stunden pro Lehrprogrammstunde realistisch.

Wenn man die Gleichung (1.3) nach der Klassenzahl n_K auflöst, also nach der Anzahl der Klassen, denen bei Direktschulung der genau gleiche Lehrstoff vermittelt werden müsste, so ergibt sich folgende sehr einleuchtende Beziehung :

Objektivierete Schulung ist rentabel, wenn

$$n_K \geq \frac{T_L}{T_D} \quad (1.4)$$

Wenn man für T_L einen Mittelwert von 120 Stunden einsetzt und für T_D die oben erwähnten Werte, so erhält man für die verschiedenen Schulstufen folgende Rentabilitätsgrenzen :

$$\text{Für Hochschul-Vorlesungen} \quad : \quad n_K \geq 20 \quad (1.5)$$

$$\text{Für Hochschul-Gruppenübungen} \quad : \quad n_K \geq 40 \quad (1.6)$$

$$\text{Für Primar- und Sekundarschule} \quad : \quad n_K \geq 60 \quad (1.7)$$

Diese Bedingungen sind natürlich auf Primar- und Sekundarstufe problemlos für alle Lehrstoffe zu erfüllen, die sich für objektivierte Schulung eignen. Weil jedoch die wenigsten Schulen über einen Computer verfügen und die Kosten pro Schülerstunde tief liegen müssen, fallen k_A und k_B (siehe Formel (1.1)) relativ stark ins Gewicht, so dass der CUU auf dieser Stufe wahrscheinlich auf viele Jahre hinaus noch keine Chance hat.

Wesentlich anders sieht die Situation auf Hochschulstufe aus. Die Kosten pro Unterrichtsstunde sind beträchtlich höher, so dass k_A und k_B weniger ins Gewicht fallen. Zudem verfügen heute alle Hochschulen bereits über Computer. An grösseren Mittelschulen könnte in einigen Jahren eine ähnliche Ausgangslage vorhanden sein.

Der Ersatz einer Vorlesung durch CUU an einer einzelnen Hochschule scheint gemäss Gleichung (1.5) in Anbetracht der mittleren Nutzungsdauer des Hochschullehrstoffs von 10 - 14 Jahren (siehe S. 5) noch nicht zweckmässig zu sein. Wenn jedoch an mehreren Hochschulen in grossen Anfängervorlesungen (z.B. Physik, Chemie, Mathematik, Ingenieurwissenschaften, Medizin usw.) das gleiche CUU-Programm eingesetzt werden könnte, so wäre die Rentabilitätsgrenze bereits überschritten. Dies würde jedoch eine gute Koordination zwischen den beteiligten Dozenten der verschiedenen Universitäten voraussetzen. Wer die Situation an Hochschulen etwas kennt, weiss jedoch, wie viele Probleme bis zu einer solchen Koordination noch zu lösen wären.

Viel günstiger ist die Lage, wenn CUU an Hochschulen in der Uebungsphase, z.B. an Stelle von Gruppenübungen zu Anfängervorlesungen, eingesetzt wird (vergl. [38]). Solche grossen Vorlesungen und die zugehörigen Uebungen werden von ca. 100 - 500 Studenten besucht, die zugehörige Gruppengrösse g beträgt dabei etwa 5 - 25 Studenten. Somit ergeben sich bei den Uebungen zu einer solchen Massenvorlesung an einer einzigen Hochschule jährlich $n_K = 4 - 100$ Uebungsgruppen, so dass die Rentabilitätsgrenze gemäss (1.6) resp. (1.4) innerhalb der Nutzungsdauer des Lehrstoffs sehr leicht überschritten werden kann.

Für die von unserem Institut betreuten Uebungen in Technischer Elektrizitätslehre III ergaben sich in den Jahren 1975 - 1977 im Mittel etwa folgende Werte :

180 Studenten pro Jahreskurs, $g = 15$ und somit $n_K = 12$.

Die Rentabilität von CUU-Uebungen ist in diesem Kurs also nach etwa 3,3 Jahren erreicht gemäss Formel (1.6) .

Je kleiner die Klassengrösse g wird, desto kleiner wird auch die für eine rentable objektivierte Schulung minimal nötige Adressatenzahl a . Im Extremfall des Einzelunterrichts, z.B. bei hochbezahlten Spezialisten, die nur zu gewissen unregelmässigen Zeiten geschult werden können, wird $g = 1$ und damit $n_K = a$.

In den letzten beiden Kapiteln wurden allgemeine wirtschaftliche Aspekte des CUU-Einsatzes behandelt. Konkrete Angaben zur Situation beim Lehrsystem PLANETH finden sich in Kap. 8. In Kap. 9 werden zum Vergleich auch die effektiven Kosten beim bekannten amerikanischen Lehrsystem PLATO IV angegeben.

1.4.3 Gegenwärtige Einsatzmöglichkeiten für CUU

In der Schweiz mit ihrem dezentralisierten Bildungswesen ist der weitverbreitete Einsatz eines teuren Grosscomputer-CUU-Systems noch auf Jahre hinaus kaum denkbar. Soll CUU trotzdem in grösserem Umfang erprobt werden, so muss dies an Institutionen geschehen, die über genügend grosse potentielle Adressatenzahlen und eventuell schon über Computer verfügen. Diese Institutionen sind die beiden technischen Hochschulen und die Universitäten, eventuell später auch Techniken und grössere Mittelschulen (vergl. [56]).

In Kap. 1.4.2 wurde gezeigt, dass der Einsatz von CUU bei Gruppenübungen zu Massenvorlesungen bereits an einer einzigen Hochschule rentabel sein kann. Dabei wurde der meist bessere Wirkungsgrad von CUU, der eher einer Privatstunde als einer Gruppenübung entspricht, noch gar nicht berücksichtigt (vergl. Kap. 1.1) .

CUU lässt sich an einer Hochschule mit Vorteil für alle Lehrformen einsetzen, die individuell durchgeführt werden müssen: Uebungen, Tests, Problemlösen, Simulationen usw. . Für den ersten Kontakt mit neuem Lehrstoff ist CUU jedoch normalerweise weniger geeignet (siehe Kap. 1.4.2). Als Vorlesungersatz wäre CUU nur einsetzbar, wenn das

gleiche Programm an vielen verschiedenen Hochschulen eingesetzt würde, wie dies in den USA zum Teil bereits geschieht. Ergänzende persönliche Kontakte mit den beteiligten Dozenten in Seminarien oder Kolloquien sind jedoch auch bei CUU-Vorlesungen unerlässlich.

Auch in der Industrie ist der Einsatz von CUU durchaus denkbar. Meist sind dort die Adressatenzahlen geringer, aber auch die Klassengrößen, so dass trotzdem noch genügend grosse Werte für n_K erreicht werden können (vergl. [23]). Da jedoch die Werte k_A und k_B (siehe Formel (1.1)) genügend klein werden müssen, kommt ein CUU-Lehrsystem vorläufig nur für Grossbetriebe oder für gemeinsame Schulungszentren mehrerer Betriebe in Frage. Für den Einsatz in der Industrie eignen sich wegen der geringeren Investitionskosten Lehrautomaten oft besser.

Es muss jedoch klar festgehalten werden, dass der CUU auch in Zukunft nur einen kleinen Teil der gesamten Unterrichtsaufgaben übernehmen kann. Der Computer wird die Lehrer also keineswegs überflüssig machen. Er kann sie aber beim Vermitteln grundlegender Kenntnisse und Fertigkeiten entlasten und ermöglicht neuartige, sonst nicht realisierbare Unterrichtsformen. Dadurch kann der Unterricht effizienter und abwechslungsreicher gestaltet werden und die Lehrer gewinnen mehr Zeit für die Entwicklung der konstruktiven und synthetischen Fähigkeiten jedes einzelnen Schülers.

2. KONZEPT UND PRINZIPIELLE FUNKTIONSWEISE DES KLEINCOMPUTERGESTEUERTEN LEHRSYSTEMS PLANETH

2.1 Kurzer historischer Rückblick

Am Mikrowellenlabor der ETH besteht schon lange grosses Interesse für neue Unterrichtsformen. Bereits seit 1969 wird in der vom Institutsleiter, Prof. Dr. G. Epprecht, gehaltenen Vorlesung "Technische Elektrizitätslehre III" die Unterrichtskontrollanlage ALMA eingesetzt [51]. Der Dozent stellt während der Vorlesung periodisch Fragen mit Auswahlantworten zum eben vorgetragenen Lehrstoff, welche die Studenten mit je 5 an jedem Platz vorhandenen Tasten beantworten können. Auf einem einfachen Anzeigegerät wird die Verteilung der eingegebenen Antworten angezeigt. Der Dozent erhält so laufend Rückmeldungen über den Lernerfolg seiner Studenten und kann bei eventuellen Unklarheiten sofort korrigierend eingreifen.

Schon früh tauchte der Wunsch auf, den Studenten unter Verwendung der in der Vorlesung eingesetzten Auswahlfragen auch ein individuelles Training zu ermöglichen. Auf diese Weise entstanden im Frühjahr 1971 je ein Projekt für einen Lehrautomaten und ein kleincomputergesteuertes Lehrsystem. Im Sommersemester 1971 wurden unter der Leitung von P. Hermann, B. von Arx und J. Fasmer im Rahmen von je drei Semesterarbeiten erste Versuche im Zusammenhang mit diesen Projekten unternommen [39, 40, 41, 42, 43, 44]. Die anschliessende Weiterentwicklung der Lehrautomatenversion wurde von U. Waber durchgeführt [49, 50]. Die Weiterentwicklung der CUU-Version wurde im Herbst 1971 von H.Häberlin und H.Müller begonnen. Bald zeigte sich, dass für den praktischen Einsatz der Anlage unbedingt ein Compiler erforderlich war, also ein Uebersetzungsprogramm, das eine relativ leicht zu erlernende Autorensprache in die Maschinensprache umwandelte (Details siehe Kap. 6.5). Die Entwicklung dieses Compilers wurde von H.Müller übernommen, während H.Häberlin sich weiter mit dem Ausbau der Systemprogramme und der Konsolen befasste.

Das CUU-Projekt erhielt den Namen PLANETH (Programmierbare LehrANlage der ETH). Der erste Versuchsbetrieb mit Studenten auf einer Lehrkonsole wurde im Januar 1972 durchgeführt. Mitte 1972 war die erste Compilerversion fertiggestellt. Im April 1973 wurde eine zweite Konsole mit wesentlich verbesserter Schaltung in Betrieb genommen.

Wegen zu geringer Zuverlässigkeit wurde die Weiterentwicklung der Lehrautomatenversion Ende 1973 aufgegeben. Im September 1974 wurden zusätzlich 4 neue Lehrkonsolen mit nochmals verbesserter und erweiterter Elektronik in Betrieb genommen, die auch für die Eingabe numerischer Konstruktivantworten ausgelegt waren. Seit diesem Zeitpunkt wurden regelmässig PLANETH-Lehrprogramme im Unterricht eingesetzt. In den Jahren 1973 - 1976 wurde zu diesen Konsolen nach verschiedenen Vorversuchen im Rahmen von Semester- und Diplomarbeiten [45, 46, 47, 48] ein beliebig fernsteuerbares Kassettentonbandgerät entwickelt, das heute einen integrierenden Bestandteil der PLANETH-Lehrkonsolen bildet.

2.2 Konzept des Lehrsystems PLANETH

Das zu entwickelnde CUU-Lehrsystem sollte für möglichst viele verschiedene Lehrstoffe verwendbar sein und sich insbesondere für den Unterricht in ingenieurwissenschaftlichen und naturwissenschaftlichen Fächern eignen. In diesen Fachgebieten ist oft auch die Darstellung von Fotografien und von komplexen, aber trotzdem leicht verständlichen Zeichnungen und Grafiken erforderlich. Deshalb musste das Lehrsystem neben Text unbedingt auch farbige Bilder darbieten können. Das System sollte interaktiv sein, d.h. nach einer Eingabe des Adressaten sollte möglichst rasch eine Reaktion des Lehrsystems erfolgen.

Ein Anschluss an das ETH-Rechenzentrum kam nicht in Frage, weil dieses über kein leistungsfähiges Time-Sharing-Betriebssystem mit für CUU geeigneten Terminals verfügte. Es stand lediglich das relativ langsam reagierende VENUS-System mit Fernschreibern oder alphanumerischen Bildschirmkonsolen zur Verfügung, das allenfalls für Programmierkurse verwendet werden konnte [9].

Die Anschaffung eines eigenen Grosscomputers für erste CUU-Versuche kam natürlich aus finanziellen Gründen nicht in Frage. Andererseits war auf einem für spezielle Mikrowellenmessungen angeschafften Kleincomputer noch freie Rechenzeit vorhanden. Aus diesen Gründen entschloss man sich, für die Steuerung des neuen CUU-Systems diesen Kleincomputer einzusetzen. Für die Speicherung des Lehrstoffs mussten deshalb wie bei anderen Kleincomputerlehrsystemen externe Bild- und Tonspeicher verwendet werden (vergl. Kap. 1.3.1).

Für die Bildspeicherung standen zunächst verschiedene Möglichkeiten zur Diskussion: Diapositive, Filmstreifen, Film oder Mikrofilmkarten (Microfiches). Es wurde jedoch beschlossen, das Lehrsystem so auszulegen, dass der Aufwand

für die Herstellung und Aenderung der Lehrprogramme möglichst klein gehalten werden konnte. Deshalb wurden als visuelle Lehrstoffträger Diapositive gewählt, die mit einfachen Mitteln leicht herzustellen und auszuwechseln sind. Um auch die Vorteile der Doppelprojektion ausnützen zu können, sollte jeder Adressatenarbeitsplatz mit zwei Diaprojektoren mit beliebigem Zugriff auf die einzelnen Bilder ausgerüstet werden (Details siehe Kap. 3.1). Diapositive haben den Nachteil, dass sie relativ viel Platz benötigen. Da jedoch selten alle Adressaten genau das gleiche Lehrprogramm gleichzeitig absolvieren wollen, genügen bei einem CUU-Lehrsystem mittlerer Grösse meist relativ wenige Lehrprogrammkopien, so dass dieser Nachteil nicht sehr schwer wiegt.

Neben Bildern sollte auf Wunsch auch Ton dargeboten werden können. Für die Speicherung dieser ergänzenden Tonkommentare erschien die Verwendung der leicht zu handhabenden Kompaktkassetten am zweckmässigsten. Da jedoch kein Tonbandgerät im Handel war, das einen beliebigen Zugriff auf einzelne auf der Kassette gespeicherte Tonsequenzen gestattete, musste ein solches Gerät selbst entwickelt werden (siehe Kap. 3.10 und 3.11).

Bisher bekannte kleinrechnergesteuerte audiovisuelle Lehrsysteme konnten als Adressatenreaktionen meist nur Auswahlantworten verarbeiten [15, 30]. Dadurch wurde die Rechenleistung des Kleincomputers jedoch nur schlecht ausgenützt. Das neue Lehrsystem sollte mehr Möglichkeiten bieten und auch komplexere Adressatenreaktionen in Form von Freiwahlantworten verarbeiten können. Da kein Massenspeicher zur Verfügung stand und der Kernspeicher des Computers relativ klein war, wäre die Analyse von aus Text bestehenden Freiwahlantworten nur in sehr beschränktem Umfang möglich gewesen. Zudem wären dazu bei den einzelnen Adressatenarbeitsplätzen zusätzliche alphanumerische Ein- und Ausgabemöglichkeiten erforderlich gewesen. Auf die Textverarbeitung wurde deshalb von vorneherein verzichtet. Dagegen erschien die Auswertung von numerischen Konstruktivantworten mit nicht allzu hohem Aufwand realisierbar, da bei den einzelnen Lehrkonsolen nur eine Anzeigeeinheit für einige Ziffern und einige zusätzliche Antworttasten benötigt wurden.

Das Lehrsystem PLANETH ist deshalb für die Verarbeitung von maximal 5 Auswahlantworten und von numerischen Freiwahlantworten mit 4-stelliger Mantisse und 2-stelligem Exponenten ausgelegt (vergl. Kap. 3.1). Für die Eingabe der Adressatenreaktionen reichen so insgesamt 16 ei-

gentliche Antworttasten aus. Zur Erleichterung der Konsolenbedienung sind in jedem Betriebszustand immer nur die gerade aktiven Tasten beleuchtet, so dass der Adressat immer genau weiss, was das Lehrsystem von ihm erwartet. Für einige häufig vorkommende Meldungen stehen einige Anzeigelampen zur Verfügung. Die Bearbeitungszeit für die einzelnen Lehrschritte kann begrenzt werden; die noch verbleibende Restzeit wird durch ein Leuchtdiodenband angezeigt. Eine ausführlichere Uebersicht über die Möglichkeiten des Dialogs zwischen Lehrkonsole und Adressat befindet sich in Kap. 3.1 .

Das den Lehrbetrieb steuernde Systemprogramm sollte möglichst vielseitige Lehrstrategien erlauben. Bei der Beurteilung der Adressatenreaktionen sollte nicht nur eine einfache Unterscheidung zwischen richtigen und falschen Antworten, sondern auch eine flexible, abgestufte Bewertung möglich sein. Neben linearen und verzweigten markoffschen Programmstrukturen sollten auch nicht-markoffsche Lehrprogramme realisierbar sein. Die Anzahl dieser nichtmarkoffschen Verzweigungsmöglichkeiten sollte jedoch in vernünftigen und noch überschaubaren Grenzen bleiben. Deshalb wurden nur nichtmarkoffsche Verzweigungen in Funktion der folgenden Grössen vorgesehen:

- der durchschnittlichen Antwortpunktzahl, d.h. der mittleren für die Beantwortung der Fragen bisher erzielten Punktzahl.
- der durchschnittlichen Zeitpunktzahl gemäss Formel (5.16), die ein Mass für die Arbeitsgeschwindigkeit des Adressaten ist.
- der durchschnittlichen Totalpunktzahl gemäss Formel (5.17), die umso höher ist, je mehr Antwortpunkte und Zeitpunkte der Adressat erzielt.

Daneben sollten auch Zufallsverzweigungen möglich sein, die nochmals eine wesentliche Erhöhung der Sicherheit vor Mogelversuchen bringen und sicherstellen, dass auch bei mehrmaligem Durcharbeiten eines bestimmten Lehrprogramms durch den gleichen Adressaten noch ein Lehreffekt auftritt.

Das Systemprogramm sollte ferner so ausgelegt werden, dass der nachträgliche Einbau weiterer nichtmarkoffscher Verzweigungen bei Bedarf leicht möglich ist. Schliesslich sollte versucht werden, mit Hilfe einiger spezieller Befehle (z.B. BILD A, BAND A, siehe Kap. 6.6) mindestens an bestimmten vom Lehrprogrammautor vorgesehenen Stellen auch eine beschränkte Form der Lernersteuerung zuzulassen.

Der verwendete Kleincomputer sollte aber nicht nur für die Steuerung des eigentlichen Lehrbetriebs eingesetzt werden. Er sollte auch zu einer eingehenden statistischen Analyse der einzelnen Lehrprogrammdurchläufe dienen und eine laufende Ueberwachung der Qualität der eingesetzten Lehrprogramme ermöglichen. Zu diesem Zweck sollten spezielle Statistikauswerteprogramme ent-

wickelt werden (Details siehe Kap. 7). Auch bei der Herstellung der Lehrprogramme konnte der Computer wertvolle Hilfe leisten. Durch die Entwicklung eines speziellen Uebersetzungsprogramms (Compilers) sollte der Aufwand für die Erstellung des Lehrprogrammcodes möglichst klein gehalten werden (Details siehe Kap. 6.5).

Um die anteiligen Kosten des Computers für die einzelnen Adressatenplätze klein halten zu können, sollten möglichst viele Lehrkonsolen an den Computer angeschlossen werden können. Wegen des relativ geringen Datenverkehrs zwischen einer einzelnen Konsole und dem Computer war es viel zu aufwendig, für jede Konsole eine eigene Datenübertragungsleitung samt zugehöriger Interfacekarte zu verwenden. Deshalb wurde nur eine einzige Datenübertragungsleitung vorgesehen, die sogenannte Party-Line, an die maximal 32 Konsolen angeschlossen werden können (Details siehe Kap. 3.2 und 3.4).

2.2.1 Diskussion des gewählten Konzeptes

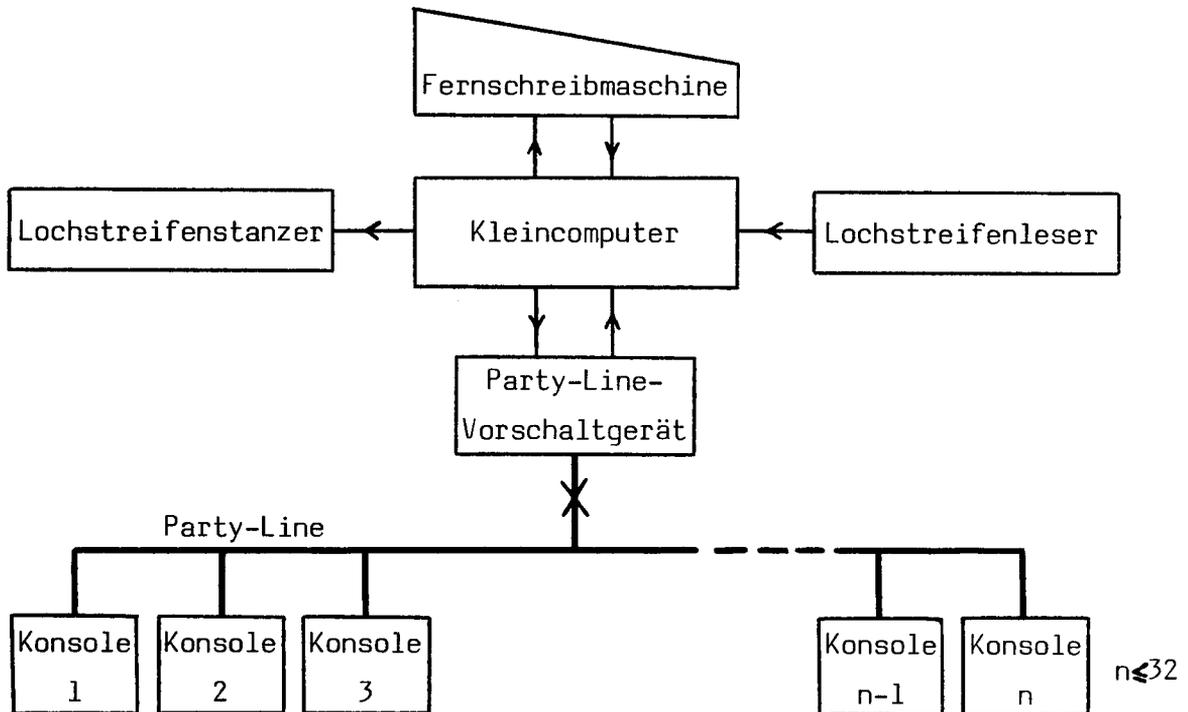
Da die visuellen Lehrinformationen fest auf Diapositiven gespeichert sind, ist die Darbietung von direkt vom Computer generierten dynamischen Bildern und Texten oder die Ueberlagerung von Fotografien mit vom Computer erzeugtem Text natürlich nicht möglich. Dies ist einer der Nachteile des gewählten Konzeptes. Da jedoch in jeder Konsole zwei Diaprojektoren statt nur einer zur Verfügung stehen, können auch beliebige Kombinationen verschiedener Bilder dargeboten werden, was die Flexibilität des Lehrsystems wesentlich erhöht und den oben erwähnten Nachteil teilweise wieder wettmacht. Da für die digitale Speicherung komplizierter farbiger Bilder sehr viel Platz benötigt wird, verwenden oft auch grosse CUU-Systeme (z.B. PLATO IV) externe, dezentralisierte Bildspeicher.

Das vorgesehene Lehrsystem arbeitet primär mit der computergesteuerten Darbietung sinnvoller Kombinationen von statischen Bildern und Tonsequenzen. Es erlaubt auch den Einsatz von farbigen Bildern, eine Möglichkeit, die viele grosscomputergesteuerte CUU-Systeme nicht besitzen und die für viele Fachgebiete (Ingenieurwissenschaften, Naturwissenschaften, Medizin usw.) unentbehrlich ist. Auch die hier realisierte Möglichkeit, Ton darzubieten, fehlt bei sehr vielen Grosscomputer-CUU-Systemen.

Ein kleincomputergesteuertes Lehrsystem kann bei der Analyse der Adressatenreaktionen natürlich nicht den gleichen Aufwand betreiben

wie ein grosscomputergesteuertes System. Gewisse Kompromisse sind unvermeidlich. Das Lehrsystem PLANETH, das Auswahlantworten und numerische Freiwahlantworten verarbeitet, stellt einen Versuch dar, die einem Kleincomputer naturgemäss gesteckten Grenzen möglichst optimal auszunützen und gewisse Nachteile bisheriger Kleincomputer-CUU-Systeme zu vermeiden.

2.3 Aufbau- und Funktionsprinzip von PLANETH



Legende: Pfeilrichtung = Signalflussrichtung

Fig. 2.1 : Blockschema des CUU-Lehrsystems PLANETH

Das Lehrsystem wird von einem Kleincomputer mit 8K 16-Bit-Worten gesteuert. Dieser Computer verfügt über eine Fernschreibmaschine, die einerseits für die Eingabe der Systemsteuerbefehle und andererseits für das Ausdrucken der Schlussprotokolle und eventueller Systemstatusmeldungen verwendet wird. Der Lochstreifenleser erlaubt das schnelle Einlesen von Systemprogramm und Lehrprogrammen (Lehrprogrammcode). Der Lochstreifenstanzer dient zum fortlaufenden Ausstanzen der während des Lehrbetriebs anfallenden Statistikdaten. Genauere Angaben über den Computer und seine Peripherie befinden sich in Kap. 4.1 und 4.2 .

Der Datenverkehr mit den angeschlossenen Lehrkonsolen geschieht über das Party-Line-Vorschaltgerät und die daran angeschlossene, allen Konsolen

gemeinsame Datenübertragungsleitung, die Party-Line. Das Vorschaltgerät dient in erster Linie zur Pegelwandlung zwischen dem niedrigen Spannungsniveau der angeschlossenen Computer-Interfacekarte und dem zwecks Erzielung einer besseren Störsicherheit höher gewählten Spannungsniveau der Party-Line-Signale. Da es den Anschluss von maximal 32 Konsolen an eine einzige Interfacekarte erlaubt, hat es gleichzeitig auch die Funktion eines Datenkonzentrators. Daneben liefert es dem Computer auch den Takt für die Systemuhr. Das Vorschaltgerät ist im Detail in Kap. 4.3 beschrieben. Auf der Party-Line werden die Daten von und zu den einzelnen Konsolen im Zeitmultiplexverfahren übertragen. Nähere Einzelheiten dazu sind in Kap. 3.2 und 3.4 zu finden.

Die Lehrkonsolen speichern die vom Computer kommenden Befehle für ihre Konsolenperipherieeinheiten und sorgen dafür, dass diese richtig ausgeführt werden. Eventuelle Tastendrucke des Adressaten werden zwischengespeichert, bis der Computer die entsprechende Information abrufen kann. Ohne Computer ist kein Lehrbetrieb möglich. Dies ist kein Nachteil, da die Zuverlässigkeit des Computers sehr hoch ist und da dank der noch überschaubaren Grösse der ganzen Anlage keine plötzlichen Systemzusammenbrüche auftreten. An jeder Konsole kann ein Adressat unabhängig von den andern Lehrkonsolen ein beliebiges Lehrprogramm absolvieren. Die Funktion der ganzen Konsole ist in Kap. 3.3, diejenige der einzelnen Baugruppen in den Kap. 3.4 bis 3.12 beschrieben.

Damit überhaupt ein Lehrbetrieb möglich ist, muss im steuernden Computer das PLANETH-Systemprogramm laufen (Details siehe Kap. 5). Dieses Programm wiederum braucht für seine Funktion die richtigen Lehrprogramme. Die Herstellung von Lehrprogrammen wird in Kap. 6 ausführlich behandelt.

Die gewählte Konfiguration von Computer und Peripheriegeräten eignet sich aber nicht nur für den eigentlichen Lehrbetrieb, sondern auch für das Uebersetzungsprogramm (SESAM-Compiler, siehe Kap. 6.5) und die Statistikauswerteprogramme (siehe Kap. 7). Alle für Erstellung, Betrieb und Auswertung benötigten Programme können beim Lehrsystem PLANETH somit auf dem gleichen Kleincomputer betrieben werden.

3. DIE PLANETH-LEHRKONSOLE

3.1 Die verschiedenen Möglichkeiten des Dialogs zwischen Konsole und Adressat

Für die Präsentation der Informationen, Fragen und Kommentare stehen in erster Linie zwei Kodak S-RA Diaprojektoren mit Rundmagazinen für 80 Dias zur Verfügung. Diese gestatten einen beliebigen Zugriff zu den einzelnen Bildern. Dank der Verwendung von zwei Projektoren (statt nur einem) können auch Kombinationen verschiedener Bilder dargestellt werden, was eine bessere Ausnutzung der einzelnen Dias und eine unabhängige Herstellung von Text- und Bildvorlagen erlaubt (keine teure Fotomontage notwendig!). Daneben können mit einem Kassettentonbandgerät, das normale Kompakt-Kassetten verwendet, bis zu 127 verschiedene Sprach- und Tonsequenzen in beliebiger Reihenfolge und unabhängig von den gezeigten Bildern abgespielt werden. Dank der Unabhängigkeit von Bild und Ton entstehen somit fast unbegrenzte Kombinationsmöglichkeiten. Eine numerische Anzeige erlaubt die Darstellung der bisher erzielten Punktzahl (ein wachsendes Punktekonto motiviert sehr stark!) oder die Kontrollausgabe eingetippter Zahlen. Ferner stehen für die Anzeige häufig vorkommender Meldungen auch einige Lampen zur Verfügung (RICHTIG, FALSCH, WARTEN, FEHLMANIPULATION und ZUM ASSISTENTEN). Die Präsentationszeit der einzelnen Bilder kann durch das Lehrprogramm begrenzt werden; die noch verbleibende Antwortzeit wird durch eine Reihe von Leuchtdioden angezeigt.

Der Adressat antwortet über eine Tastatur mit 16 Tasten (Zahlen von 0 bis 9, - , Komma, START, STOP, WEITER und ZUSATZZEIT). Dabei sind in jedem Programmzustand immer nur jene Tasten beleuchtet, die Antworten entgegennehmen. Neben diesen eigentlichen Antworttasten sind noch drei Hilfstasten vorhanden (TONREPETITION, BILDSTOERUNG und TONSTOERUNG). Je nach Fragestellung wählt der Student eine von fünf vorgegebenen Auswahlantworten oder er gibt eine von ihm berechnete reelle Zahl ein. Diese darf eine maximal 4-stellige Mantisse und einen 2-stelligen Exponenten besitzen und muss betragsmässig zwischen 10^{-38} und 10^{+38} (oder bei 0) liegen. Die numerischen Konstruktivantworten erweitern die Einsatzmöglichkeiten enorm. Das System kann eine Zahlenantwort nach der Grösse des Fehlers klassieren und ist auch in der Lage, Komma- und Vorzeichenfehler zu erkennen und zu bewerten.

Die Bildmagazine und die Tonkassetten sind leicht zugänglich und können von den Studenten selbst ausgewechselt werden. Fig. 3.1 zeigt die Frontplatte einer PLANETH-Konsole, Fig. 3.2 die beiden in die Konsole eingebauten Projektoren mit den eingelegten Magazinen.

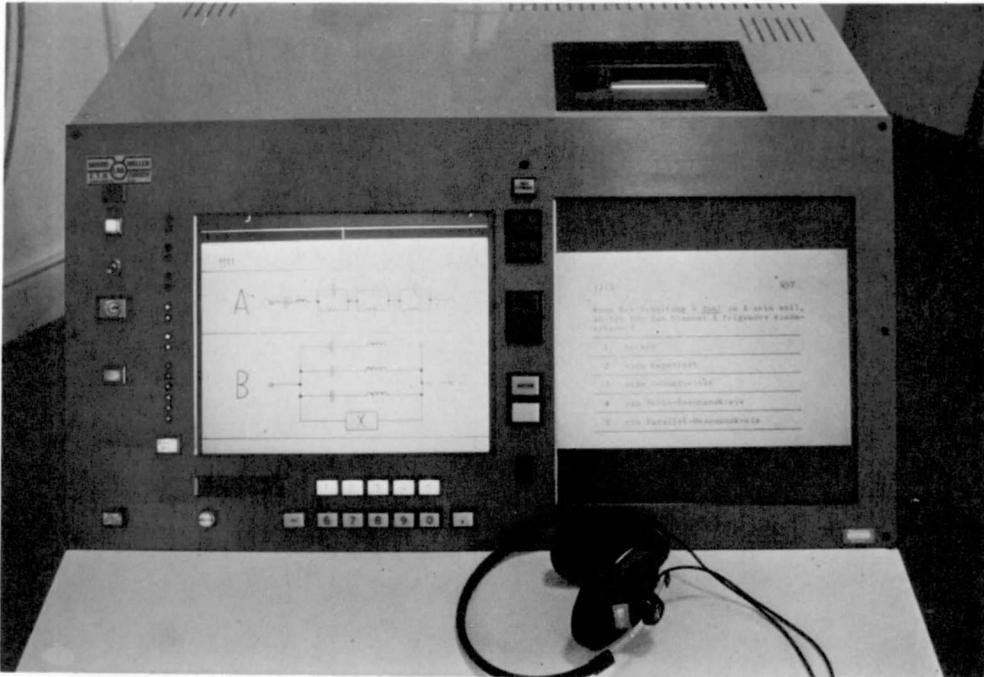


Fig. 3.1 : Frontansicht einer PLANETH-Konsole.
Die Tonkassetten werden in den Schlitz bei der
Vertiefung rechts oben eingeschoben.

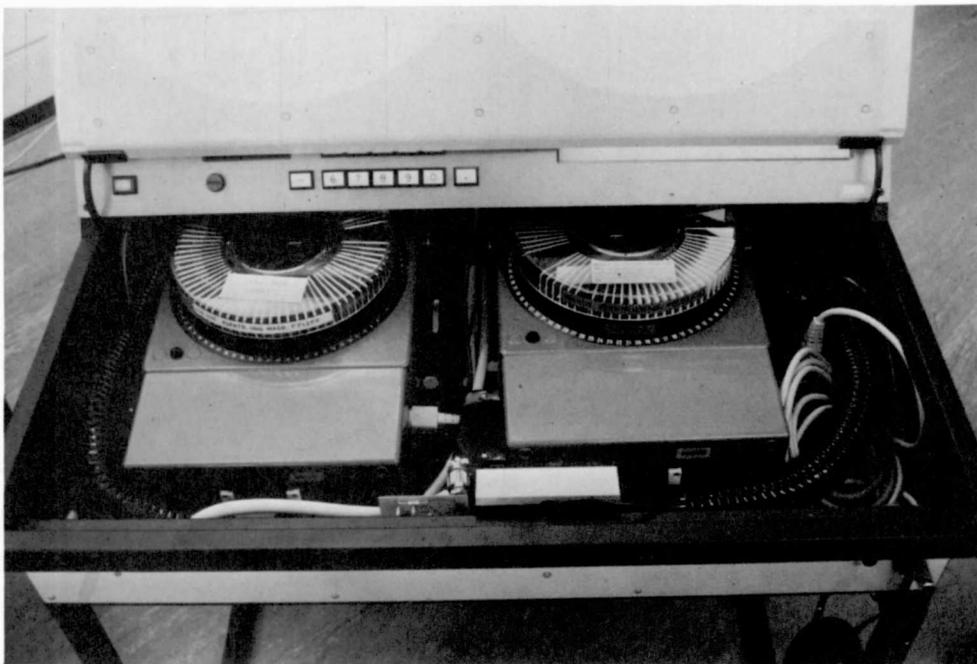


Fig. 3.2 : Blick auf die beiden in die Konsole eingebauten Bildprojektoren. Der Pultdeckel ist in Magazinwechselstellung.

3.2 Das Prinzip der Datenübertragung zwischen Computer und Konsole

Der Speicher des Hewlett-Packard Kleincomputers, der das Lehrsystem steuert, ist in 16-Bit-Worten organisiert. Es ist also zweckmässig, für die Adressierung und Datenübertragung zusammen 16 Bits zu verwenden. Die 7 Einheiten der Konsolenperipherie (2 Projektoren, Anzeigelampen, Zeitanzeige, numerische Anzeige, Tonbandgerät, Tastatur) benötigen maximal 8 Bits für die Datenübertragung und 3 Bits für ihre interne Adressierung. Es verbleiben somit noch 5 Bits, die die Adressierung von maximal 32 Konsolen gestatten. Die 16 Bits eines Computerwortes werden für die Datenübertragung wie folgt aufgeteilt:

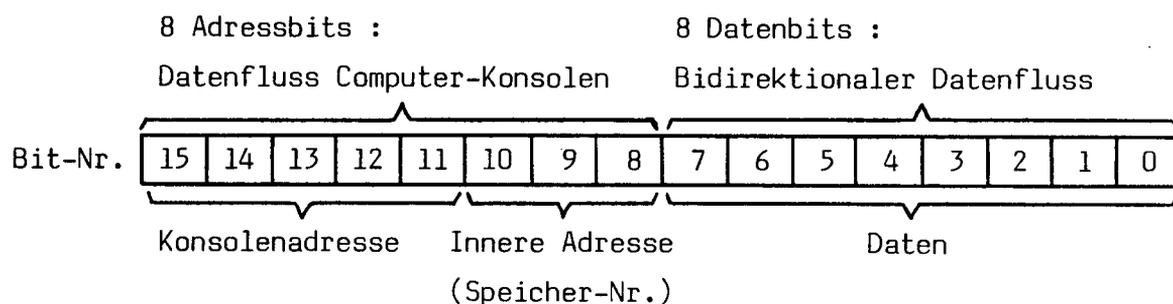


Fig. 3.3 : Die Aufteilung der Party-Line-Bits für die Datenübertragung

Diese 16 Bits werden jeweils parallel auf eine allen Konsolen gemeinsame Datenübertragungsleitung (Party-Line) ausgegeben. Auf den 8 Adressbitleitungen fließen die Signale immer vom Computer zu den Konsolen. Der Computer steuert damit den Informationsfluss auf den 8 Datenbitleitungen, der in beiden Richtungen erfolgen kann. Ein weiteres Signal, der Systemtakt (Control), steuert den Zeitpunkt der Datenübergabe. Daneben sind auf der Party-Line noch 3 Bits Statusinformationen vorhanden.

Da der Computer 32 Konsolen bedienen muss, kann er sich nur eine ganz kurze Zeit mit einer bestimmten Konsole befassen. Die ankommenden Daten für jede Einheit der Konsolenperipherie müssen also jeweils sofort gespeichert werden. Ebenso müssen die von der Tastatur kommenden Daten gespeichert werden, bis sie der Computer übernehmen kann. Der Computer ruft bei jeder Konsole die Information im Tastenspeicher periodisch ab. Wenn eine Taste gedrückt wurde oder wenn die Konsole aus andern Gründen neue Information braucht, veranlasst er unmittelbar darauf die Ausgabe der benötigten Daten an die angewählte Konsole. Infolge dieser zyklischen Abfrage der einzelnen Konsolen erfolgt die Datenübertragung auf der Party-Line also in einer Art Zeitmultiplex [1]. Im Gegensatz zum normalen Zeitmultiplex-Verfahren ist hier

aber das Zeitintervall, in dem der Informationsaustausch mit einer Konsole stattfindet, nicht konstant, sondern es ist abhängig von der Menge der auszutauschenden Daten.

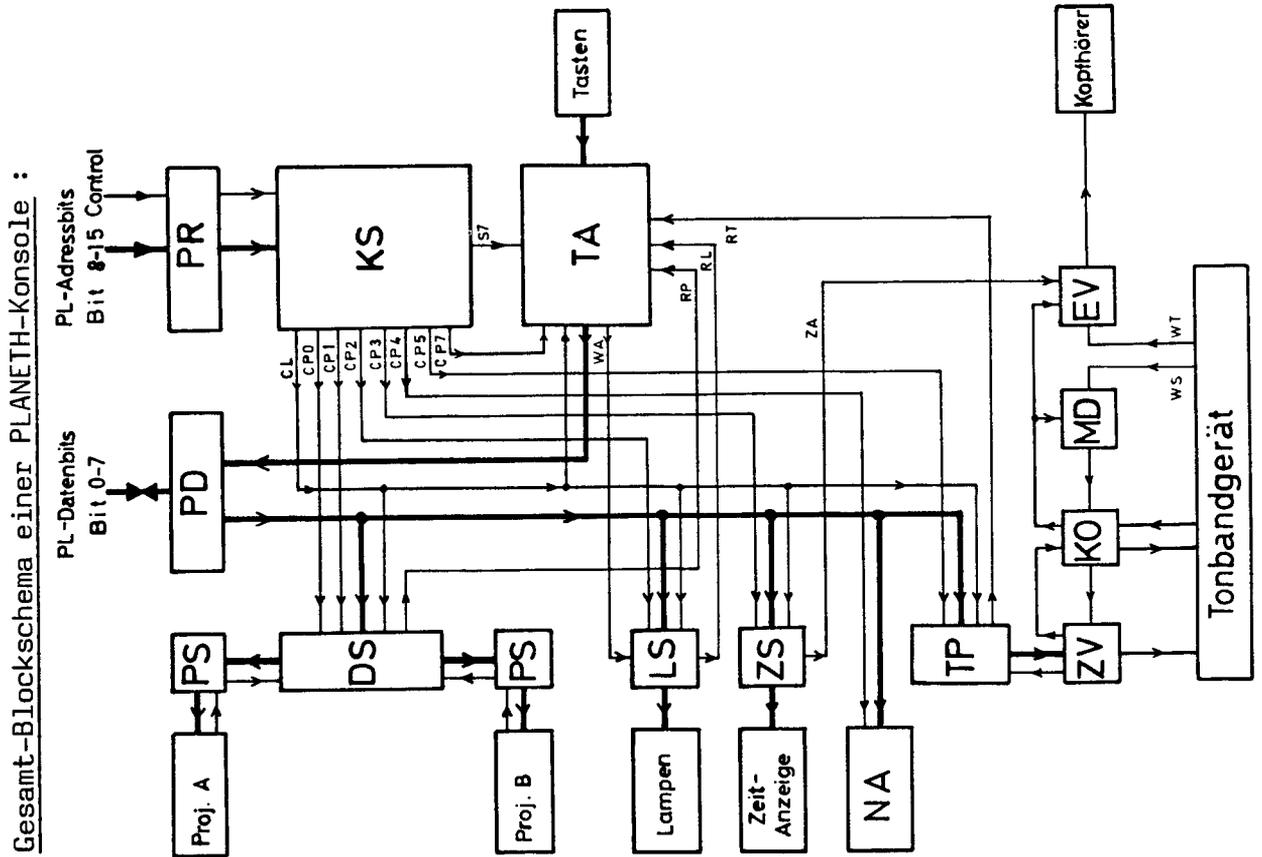
Die Signalpegel auf der Computerseite sind TTL-kompatibel (ground-true, logisch 1 \approx 0V, logisch 0 \approx 3V). Sie sind also für den direkten Anschluss einer unter Umständen mehrere hundert Meter langen Leitung viel zu niedrig. Ueberdies ist die TTL-Logik für den direkten Anschluss langer Leitungen viel zu schnell. Deshalb wurden für die Party-Line Signalpegel von 0V (logisch 1) und +24V (logisch 0) gewählt. Die Eingänge der Party-Line-Empfangsschaltungen wurden absichtlich hochohmig und relativ langsam ausgelegt, um die Datenübertragung unempfindlich gegen schnelle Störungen zu machen. Die Sendeschaltungen sind so dimensioniert, dass die Signale eher rampenförmig als schrittweise ansteigen und abfallen, was das Uebersprechen zwischen den einzelnen Bitleitungen ebenfalls verringert. Dank diesen Massnahmen wurde eine hohe Störsicherheit erreicht, welche die Verwendung von fehlerkorrigierenden Spezialcodes [1] für die Datenübertragung überflüssig machte.

Eine genauere Beschreibung der Signalübertragung und der Sende- und Empfangsschaltungen befindet sich in Kap. 3.4 .

3.3 Gesamt-Blockschema und prinzipielle Funktionsweise der Konsole

Fig. 3.4 zeigt das Gesamt-Blockschema einer PLANETH-Konsole. Darin sind nur die wichtigsten Signale zwischen den einzelnen Baugruppen eingezeichnet.

Die Konsolenelektronik ist zu einem grossen Teil aus TTL-Logik-Bau-elementen aufgebaut. Die Baugruppen PR und PD besorgen die Pegelwandlung zwischen den Party-Line-Signalen ("0"=24V, "1"=0V) und den internen TTL-Signalen ("0"=0V, "1"=3V). In PR sind die Empfangsschaltungen der 8 Adressbits (Bit 15-8) und des Systemtakts, in PD die Sendeschaltungen (für Konsolenoutput) und die Empfangsschaltungen (für Konsoleninput) der 8 Datenbits (Bits 7-0).



Gesamt-Blockschema einer PLANETH-Konsole :

- Legende :
- Baugruppen :
- DS Dianummern-Speicher (Speicher 0, 1)
 - EV NF-Endverstärker
 - KO Korrekturstufe
 - KS Konsolensteuerereinheit
 - LS Lampensteuerung (Speicher 2)
 - MD Markendetektor
 - NA Numerische Anzeige (Speicher 4)
 - PD Party-Line-Driver/Receiver (Pegelwandler)
 - PL Party-Line
 - PR Party-Line-Receiver (Pegelwandler)
 - PS Projektorsteuerung
 - TA Tastenspeicher (Speicher 7)
 - TP Tonband-Positions-Speicher (Speicher 5)
 - ZS Zeitanzeige-Steuerung (Speicher 3)
 - ZV Zähler-Vergleicher
- Signale :
- CL Einschalt-Clear
 - CP0 Takt Speicher 0 (Projektor A)
 - CP1 Takt Speicher 1 (Projektor B)
 - CP2 Takt Lampenspeicher (2)
 - CP3 Takt Zeitanzeigespeicher (3)
 - CP4 Takt Num.-Anzeige-Speicher (4)
 - CP5 Takt Tonband-Positions-Speicher (5)
 - CP7 Takt Tastenspeicher (7)
 - RL Ready Lampen
 - RP Ready Projektoren
 - RT Ready Tonband
 - S7 Steuersignal Tastenspeicher-Ausgangstore
 - WA Warten (Tasten gesperrt)
 - WS Wiedergabesignal Steuerspur
 - WT Wiedergabesignal Tonspur
 - ZA Zeitablauf (für akustische Warnung)

Fig. 3.4 : Gesamt-Blockschema einer PLANETH-Konsole

Die Konsolensteuerung KS kontrolliert den Datenfluss in der ganzen Konsole. Sie bildet aus den 8 Adressbits und dem Systemtakt (Control) die Takte (CPO - CP7) für die Speicher der Konsolenperipherieeinheiten. Diese Speicher übernehmen ihre Daten, wenn das entsprechende Taktsignal von 0 auf 1 geht. Dies ist nur dann der Fall, wenn die richtige Konsolenadresse und die richtige innere Adresse bei KS vorhanden ist und wenn der Systemtakt von 0 auf 1 geht. Unmittelbar nach dem Einschalten der Konsole gibt KS jeweils ein Signal CL (Einschalt-Clear = Löschimpuls nach Einschalten) aus, das alle Konsolenspeicher löscht, damit keine undefinierten Zustände auftreten.

Die Baugruppe DS enthält je einen Speicher für die von der Eingangsdatensammelschiene kommenden Dianummern der Projektoren A und B. Die im natürlichen Binärcode vorliegenden Zahlen werden in PS für die Ansteuerung der beiden Kodak S-RA Projektoren umcodiert. Wenn eine neue Dianummer gespeichert worden ist, läuft der entsprechende Projektor jeweils sofort auf die neue Position. Während er in Bewegung ist, ist das Rückmeldesignal RP auf 0 und blockiert die Eingabetastatur. Sobald beide Projektoren stillstehen und die gewünschten Dias zeigen, geht RP auf 1 und gibt die Tasten wieder frei.

In LS werden die Lampensteuerbefehle gespeichert, decodiert und über Treiberschaltungen die entsprechenden Lampen angesteuert. Neben den Meldungslampen RICHTIG, FALSCH usw. steuert LS auch die Tastenlampen, die anzeigen, welche Tasten gerade aktiv sind. Falls die Tasten vom Computer aus gesperrt werden sollen, setzt ein spezieller Lampenbefehl das Signal RL auf 0. Wenn die Tasten gesperrt sind, gibt die Baugruppe TA ein Signal WA=1 aus, das in LS das Anzünden der WARTEN-Lampe veranlasst.

Die Zeitsteuerung ZS speichert die anzuzeigende Restzeit in binärer Form und erzeugt daraus die Ansteuersignale für die einzelnen Minuten- und Sekunden-Anzeige-Leuchtdioden. Wenn die Restzeit auf weniger als 10 Sekunden gesunken ist, wird die Baugruppe EV durch das in ZS generierte Signal ZA veranlasst, eine akustische Warnung in den Tonkanal einzublenden.

Die numerische Anzeigeeinheit NA besteht aus 8 Ziffern. Jede kann die Zahlen 0 - 9, das Minuszeichen sowie nichts (BLANK) anzeigen. Unabhängig davon kann bei jeder Ziffer noch ein Dezimalpunkt (Komma) angesteuert werden. Von den 8 Datenbits werden für die Adressierung dieser 8 Ziffern 3 Bits benötigt, für den Dezimalpunkt 1 Bit und für

den Zahlenwert 4 Bits. Die verwendeten Ziffernanzeigen haben einen eingebauten Speicher, so dass in NA nur noch aus CP4 und den 3 Ziffernadressbits die Taktpulse für die internen Zifferspeicher erzeugt werden müssen.

Die Tonbandsteuerung besteht aus insgesamt 5 Baugruppen, nämlich TP, ZV, KO, MD und EV. Mit Ausnahme von TP sind sie zur Erhöhung der Störsicherheit aus CMOS-Logik aufgebaut, die mit 12V gespeist wird. Das angeschlossene 2-Spur Stereo-Tonbandgerät wurde speziell für die PLANETH-Konsolen entwickelt. Alle seine Funktionen sind mit CMOS-kompatiblen Signalen fernsteuerbar. Auf einer der beiden Spuren, der Tonspur, sind die einzelnen Sprach- und Tonsequenzen aufgenommen. Auf der andern Spur, der Steuerspur, ist jede Sequenz mit je einem niederfrequenten (25Hz) und einem hochfrequenten (3,2kHz) Tonimpuls markiert. Wenn eine neue Tonsequenz abgespielt werden soll, so wird die Nummer der gewünschten Sequenz zunächst in TP gespeichert. Wenn diese Nummer nicht mit der aktuellen Tonbandposition übereinstimmt, die in einem Zähler in ZV enthalten ist, gibt ZV einen Vor- oder Rücklaufbefehl an das Tonbandgerät aus. Die niederfrequenten Tonimpulse können auch im schnellen Suchlauf noch gelesen werden. MD bildet genau einen CMOS-Zählimpuls pro NF-Tonimpuls. Dieser Zählimpuls gelangt über die Korrekturstufe KO in den Positionszähler in ZV. TP setzt während des Suchvorgangs das Rückmeldesignal RT auf 0 und blockiert so die Eingabetasten.

Sobald die Vergleichsstufe in ZV Gleichheit von Soll- und Istposition feststellt, wird das Tonband gebremst und KO veranlasst den Abspielvorgang. Das Signal der Tonspur wird in EV verstärkt und darauf dem Kopfhörer zugeführt. Der Schluss jeder Tonsequenz ist mit einem hochfrequenten Tonimpuls markiert. Sobald MD diesen Impuls auf der Steuerspur feststellt, wird das Tonbandgerät via KO gestoppt. Eine genauere Funktionsbeschreibung der Tonbandsteuerung befindet sich in Kap. 3.11 .

Ein Signal von einer Antworttaste wird in TA zunächst gefiltert und entstört. Anschliessend wird es für die Uebertragung codiert und gespeichert. Darauf bleiben die Tasten solange gesperrt, bis die Information im Tastenspeicher TA vom Computer übernommen worden ist. Die Tasten können auch durch die Rückmeldungen RP, RL oder RT (von den Baugruppen DS, LS resp. TP) blockiert werden. Sobald KS merkt, dass der Computer die Information des Tastenspeichers einlesen will, geht das Steuersignal S7 auf 1. Dadurch werden die Ausgangstore des TA-Speichers geöffnet und die Tasteninformation gelangt über PD auf die Party-Line-Datenbitleitungen. Wenn der Takt CP7 von 1 wieder auf 0 geht, wird der TA-Speicher gelöscht und ist für die Aufnahme neuer Information bereit.

Die ganze Konsolenelektronik ist modular aufgebaut. Deshalb wäre es auch ohne weiteres möglich, durch Weglassen nicht benötigter Baugruppen eine viel einfachere und billigere, dafür aber weniger leistungsfähige Konsole zu realisieren. Aus je einer Baugruppe PR, PD, KS, DS, PS und TA sowie einem Projektor liesse sich beispielsweise bereits eine einfache PLANETH-Konsole für die Bearbeitung von Lehrprogrammen mit Auswahlantworten zusammenstellen.

Die vorstehende Erklärung der Funktion der PLANETH-Konsole ist natürlich noch ziemlich knapp. Genauere Angaben sind in den Beschreibungen der einzelnen Baugruppen in den folgenden Kapiteln zu finden.

3.4 Die Party-Line und die dazugehörigen Sende- und Empfangs- Schaltungen

3.4.1 Prinzipielle Schaltung der Party-Line

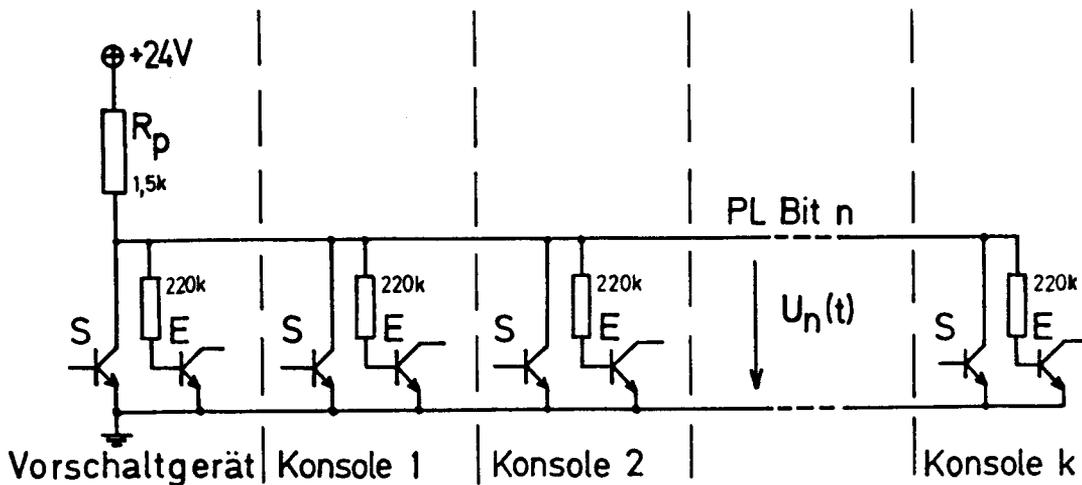
Auf der Party-Line (PL) werden die Signale zwischen dem Computer und den Konsolen übertragen. Die Schaltung dieser Datenübertragungsleitung soll möglichst wenig Einschränkungen unterliegen. Die einzelnen Konsolen sollen an beliebigen Stellen angeschlossen werden können und die ganze Leitung soll sich bei Bedarf auch problemlos in verschiedene Aeste aufspalten lassen. Da gleichzeitig die zu übertragende Datenmenge pro Zeiteinheit ziemlich klein ist, eignet sich eine nicht angepasste, relativ langsame Leitung für diese Anwendung sehr gut.

Die Party-Line ist für eine Kabellänge $l_{CK} \leq 100\text{m}$ zwischen Computer und Konsole ausgelegt. Die Laufzeit der Signale zwischen Computer und Konsole wird durch l_{CK} bestimmt und ist somit kleiner als etwa $\frac{1}{2}\mu\text{s}$, und spätestens nach einigen Mikrosekunden sind die durch Reflexion an den nichtangepassten Enden der Leitung entstandenen Wellen abgeklungen. Wenn die Party-Line aus mehreren Aesten besteht, kann l_{tot} , die totale Länge des ganzen Leitungssystems (=Summe der Längen der einzelnen Aeste), noch wesentlich grösser als l_{CK} sein. l_{tot} bestimmt die Form (Anstiegs- und Abfallzeit) der Signale auf der Party-Line und damit die maximale Geschwindigkeit der Datenübertragung (siehe Kap. 3.4.2 und 3.4.3) .

Die Adressbit-, Datenbit- und Systemtakt-Signale werden über je ein verdrehtes Aderpaar in einem 20-paarigen Computerkabel übertragen.

Eine der beiden Adern jedes Paares ist zur Verringerung der gegenseitigen Störungen geerdet. Die andere Ader ist im Vorschaltgerät, das beim Computer die Umwandlung zwischen den TTL-Pegeln und den PL-Pegeln besorgt, jeweils über einen $1,5k\Omega$ Widerstand (R_p) mit +24V verbunden. Wenn ein Signal übertragen werden soll, so schaltet auf der Sendeseite ein Transistor die Ader auf Erde. Bei den Empfangsschaltungen gelangt das PL-Signal im Prinzip über einen $220k\Omega$ Widerstand auf die Basis eines Darlington-Transistors in Emitterschaltung, an dessen Kollektorwiderstand das entsprechende TTL-Signal abgenommen werden kann.

Die Schaltung einer solchen Ader für die Uebertragung eines PL-Datenbits sieht also etwa wie folgt aus :



S = Sendeschaltungen (fehlen in den Konsolen bei Adressbits/Systemtakt)
 E = Empfangsschaltungen

Fig. 3.5 : Prinzipielle Schaltung eines PL-Aderpaares für die Uebertragung eines Datenbits

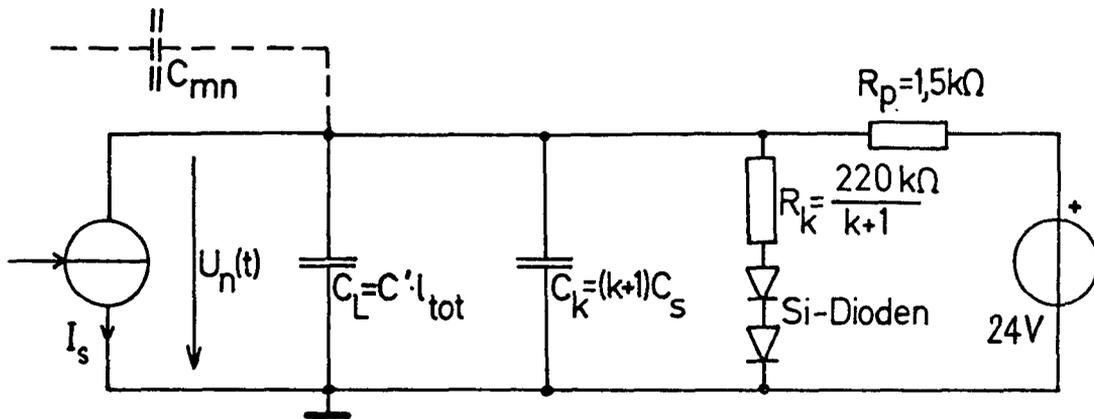
Die Empfangsschaltungen werden nun so ausgelegt, dass ihre Ansprechzeit länger ist als die Abklingzeit der Wellen, die bei Signaländerungen infolge der Fehlanpassungen an den Leitungsenden entstehen. Der Widerstandsbelag R' eines Aderpaares beträgt ca. $0,2\Omega/m$, so dass gilt :

$$R' \cdot l_{CK} \leq 20\Omega \ll R_p = 1,5k\Omega \quad (3.1)$$

Der Leitungswiderstand eines Aderpaares kann somit gegen den Pull-up-Widerstand $R_p = 1,5k\Omega$ vernachlässigt werden. Die Sendeschaltungen in den Konsolen werden vom Computer über die Adressbits so gesteuert, dass immer höchstens eine durchgeschaltet ist.

3.4.2 Die Ersatzschaltung der Party-Line und die Sende-/Empfangs-Schaltungen

Für die Bestimmung der Spannung $U_n(t)$ einer PL-Bit-Ader genügt in erster Näherung folgendes Ersatzschema :



- Legende:
- n Index des PL-Bits
 - k Anzahl angeschlossene Konsolen
 - C' Kapazitätsbelag eines Aderpaares (ca. 120pF/m)
 - C_L Leitungskapazität eines Aderpaares
 - C_K Aequivalente Ausgangskapazität bei k Konsolen
 - C_s Ausgangskapazität eines Sendetransistors
 - C_{mn} Kopplungskapazität zwischen den Bit-Leitungen m und n
 - I_s Strom des Sendetransistors, ca. 70 - 230mA
(nur beim Senden einer 1 eingeschaltet)
 - l_{tot} Totale PL-Länge
 - R_k Aequivalenter Eingangswiderstand der Empfangsschaltungen bei k Konsolen
 - U_n Spannung der PL-Bitleitung n ($U_n \geq 0$)

Fig. 3.6 : Ersatzschaltung der Party-Line (für 1 Bit)

Bei maximalem Ausbau mit $k = 32$ Konsolen beträgt U_n im eingeschwungenen Zustand für logisch 0 statt +24V nur noch ca. +19,8V. Der Idealwert für das Umschaltniveau zwischen 0 und 1 wäre also 9,9V. Bei den realisierten Empfangsschaltungen liegt es wegen der Verwendung von Normwiderstandswerten etwas darunter, nämlich bei etwa 9,8V. Im Detail sehen die Sende- und Empfangsschaltungen der Party-Line für 1 Bit wie folgt aus :

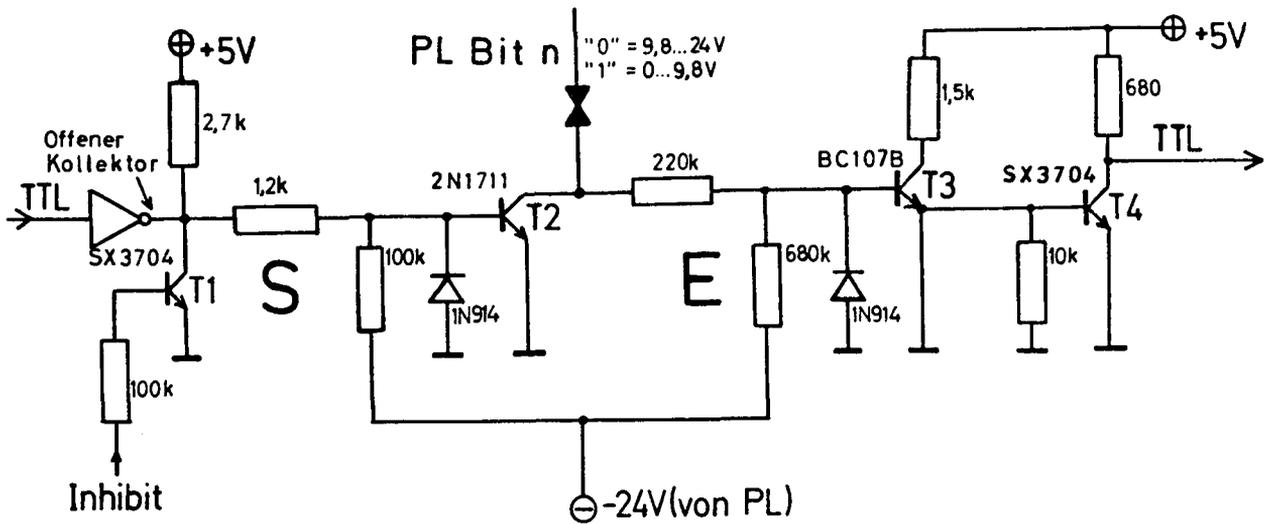


Fig. 3.7 : Schema der Party-Line Sende- und Empfangsschaltungen

Der typische statische Störabstand für die gewählte Dimensionierung liegt bei Berücksichtigung der Spannungsabfälle der Leitung und der Sättigungsspannung der Sendetransistoren für 1 etwa bei 9,5V und für 0 etwa bei 10V. Der dynamische Störabstand liegt noch etwas höher: Bei einer Frequenz von 1Mhz wurde für 1 im Mittel etwa 11V und für 0 etwa 11,5V gemessen. Die Minimalwerte für den statischen Störabstand betragen für 1 immerhin 7,3V und für 0 7,6V. Messungen an mehreren Sende-/Empfangsschaltungen zeigten jedoch meist nur sehr geringe Abweichungen von den typischen Werten.

Damit die Leckströme der maximal 33 Sendetransistoren T2 pro Bitleitung für Spannungen bis etwa 60V vernachlässigbar bleiben, erhalten ihre Basen im abgeschalteten Zustand eine leicht negative Vorspannung. T1 verhindert die Ausgabe undefinierter Daten auf die Party-Line im Moment des Ein- und Ausschaltens einer Konsole. Zu diesem Zweck erzeugt eine von der Party-Line gespeiste Hilfsschaltung in jeder Konsole ein Inhibit-Signal, das T1 durchschaltet, sobald die interne 5V-TTL-Speisespannung unter etwa 4,5V absinkt.

3.4.3 Der zeitliche Verlauf der Party-Line-Signale

Mit Hilfe der Ersatzschaltung von Fig. 3.6 lässt sich auch der Verlauf der PL-Signale in Funktion der Zeit berechnen. Die Dauer der einzelnen Impulse wird in erster Linie durch das Computerprogramm bestimmt. Ihre Anstiegs- und Abfallzeiten hingegen sind proportional zur Leitungslänge.

Wenn in Fig. 3.6 die Quelle I_s durch einen Kurzschluss ersetzt wird, so fließt ein Strom von 16mA. Da der Sendetransistor im geschlossenen Zustand die Spannung U_n möglichst genau auf 0V bringen soll, muss sein Quellenstrom I_s in jedem Fall wesentlich grösser als diese 16mA sein. Die Abfallzeit T_f (Uebergang 0→1) eines PL-Impulses wird daher immer beträchtlich kürzer sein als die Anstiegszeit T_r (Uebergang 1→0, siehe Fig. 3.8). Da $C_k \ll C_L$ und $R_k \gg R_p$ sind, können C_k und R_k für die Berechnung von T_r in erster Näherung vernachlässigt werden, d.h. es gilt :

$$T_r \approx R_p \cdot C_L \gg T_f \quad (3.2)$$

Die maximale Geschwindigkeit der Datenübertragung wird also durch T_r bestimmt. Damit der effektive Störabstand bei der Datenübertragung etwa gleich dem statischen Störabstand wird, sollten die Adress- und Datenbit-Signale etwa um T_d vor dem Systemtakt (Control) auf die Leitung gegeben werden und erst um T_d nach Verschwinden des Systemtakts wieder abgeschaltet werden, wobei :

$$T_d \geq 5T_r \quad (3.3)$$

Man könnte zwar glauben, wegen $T_f \ll T_r$ müssten die Signale vor dem Erscheinen des Systemtaktimpulses auch viel weniger lang vorhanden sein als nach seinem Verschwinden. Dies wäre nur dann richtig, wenn alle PL-Signale vor der Ausgabe der Information sicher 0 wären. Es können jedoch mehrere Datenausgabezyklen unmittelbar aufeinander folgen. Das Systemprogramm braucht für die Vorbereitung einer Datenausgabe etwa 25 μ s. Optimale Störsicherheit bei minimalem Zeitaufwand erhält man also, wenn die Signale um

$$T_v = T_d - 25\mu s \quad (3.4)$$

vor dem Systemtaktimpuls auf die Leitung gegeben werden.

Die Party-Line-Signale in Funktion der Zeit für einen Datenübertragungszyklus bei $t_{tot} = 100m$

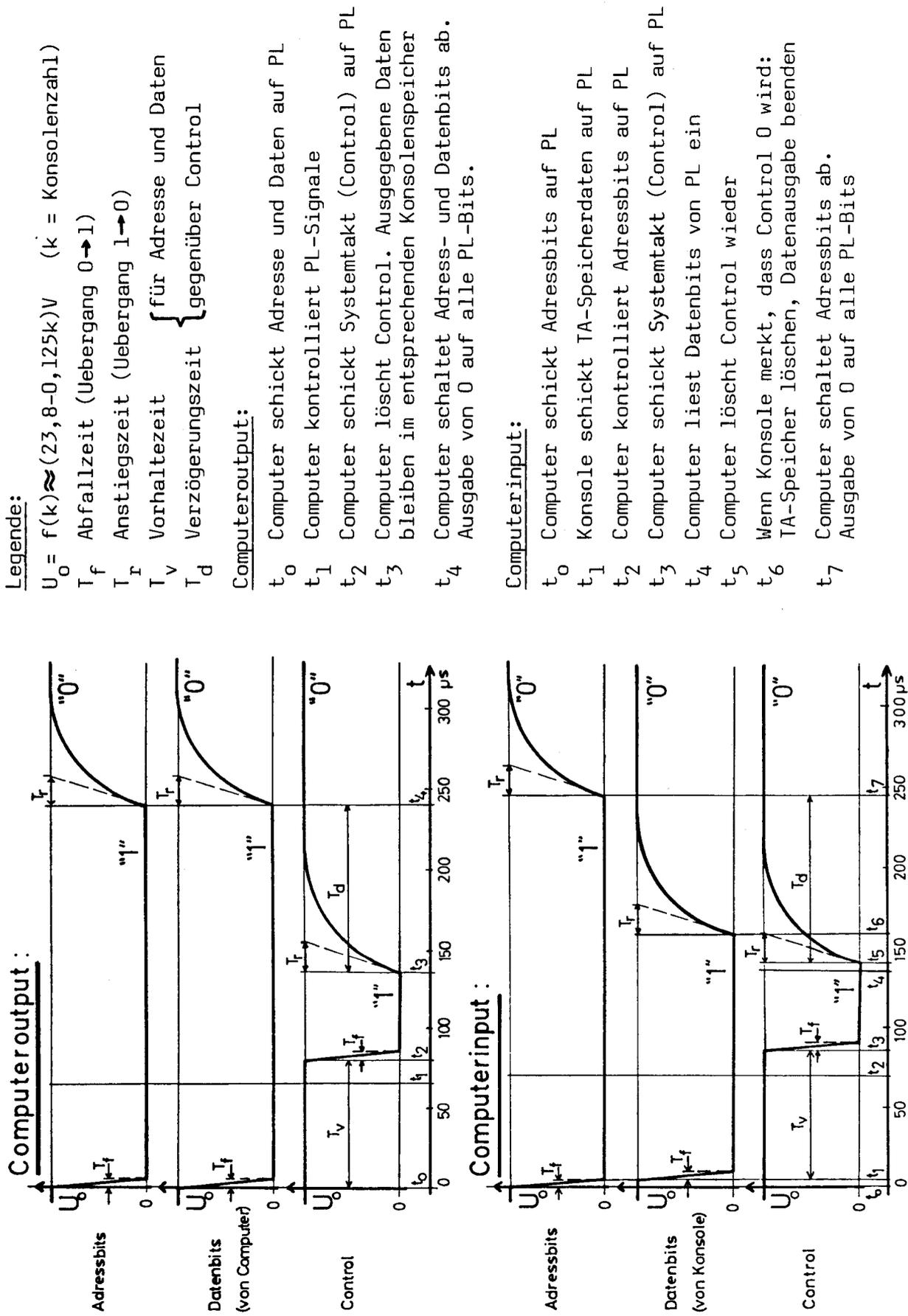


Fig. 3.8 : Die Party-Line-Signale in Funktion der Zeit für einen Datenübertragungszyklus bei $t_{tot} = 100m$

Da die Systemtakt-Leitung sicher auf 0 ist vor dem Einsetzen eines Signals, kann die Dauer eines Systemtakt-Impulses wesentlich kürzer sein. Die minimal nötige Impulsdauer dürfte etwa bei T_r liegen.

Fig. 3.8 zeigt die PL-Signale in Funktion der Zeit bei einer totalen PL-Länge $l_{tot} = 100m$ (Uebersprecheffekte weggelassen).

3.4.4 Der Einfluss des Uebersprechens

Natürlich tritt auf der Party-Line bei Signaländerungen ein gewisses Uebersprechen zwischen den einzelnen Adern auf. Speziell die kapazitiv eingekoppelten Störspannungen dauern länger und klingen langsamer ab als die infolge Fehlanpassung der Leitung entstehenden Einschwingvorgänge und können deshalb nicht grundsätzlich vernachlässigt werden. Besonders wenn viele Leitungen von 0 auf 1 wechseln, können die Störspannungen auf andern Leitungen mit Signal 0 (hochohmiger Zustand) in der Größenordnung von einigen Volt liegen. Wie die Nutzsignale klingen diese Störsignale mit der Zeitkonstanten T_r ab. Bis jedoch der Systemtakt erscheint, sind sie verschwunden, denn die Daten wurden mehrere T_r vorher auf die Leitung gegeben. Der Systemtakt als einzelnes Signal erzeugt nur ein ganz geringes Uebersprechen auf die andern 16 Leitungen. Dagegen könnte das Uebersprechen dieser Leitungen auf den Systemtakt unter Umständen zu falschen Impulsen führen. Solche Uebersprechstörungen auf den Systemtakt werden durch ein zusätzliches Tiefpassfilter und einen Schmitt-Trigger in der Baugruppe KS (Konsolensteuerung) unterdrückt.

Bei genügender Verzögerung des Systemtakts gegenüber den Adress- und Datensignalen kann das Uebersprechen somit keine Uebertragungsfehler bewirken.

3.5 Die Konsolensteuereinheit (KS)

Die Hauptaufgabe der Konsolensteuerung ist die Erzeugung der Taktimpulse (CP0 - CP7) für die internen Konsolenspeicher aus den 8 Adressbitsignalen und dem Systemtakt (Control). Ein solcher Taktpuls darf nur dann ausgegeben werden, wenn folgende Voraussetzungen erfüllt sind :

- a) Die Konsolenadresse stimmt
- b) Die innere Adresse des Speichers ist richtig
- c) Der Systemtaktimpuls (Control) ist vorhanden (d.h. es liegen am Speichereingang bereits die richtigen Daten vor).

Das Blockschema der Konsolensteuerung sieht wie folgt aus :

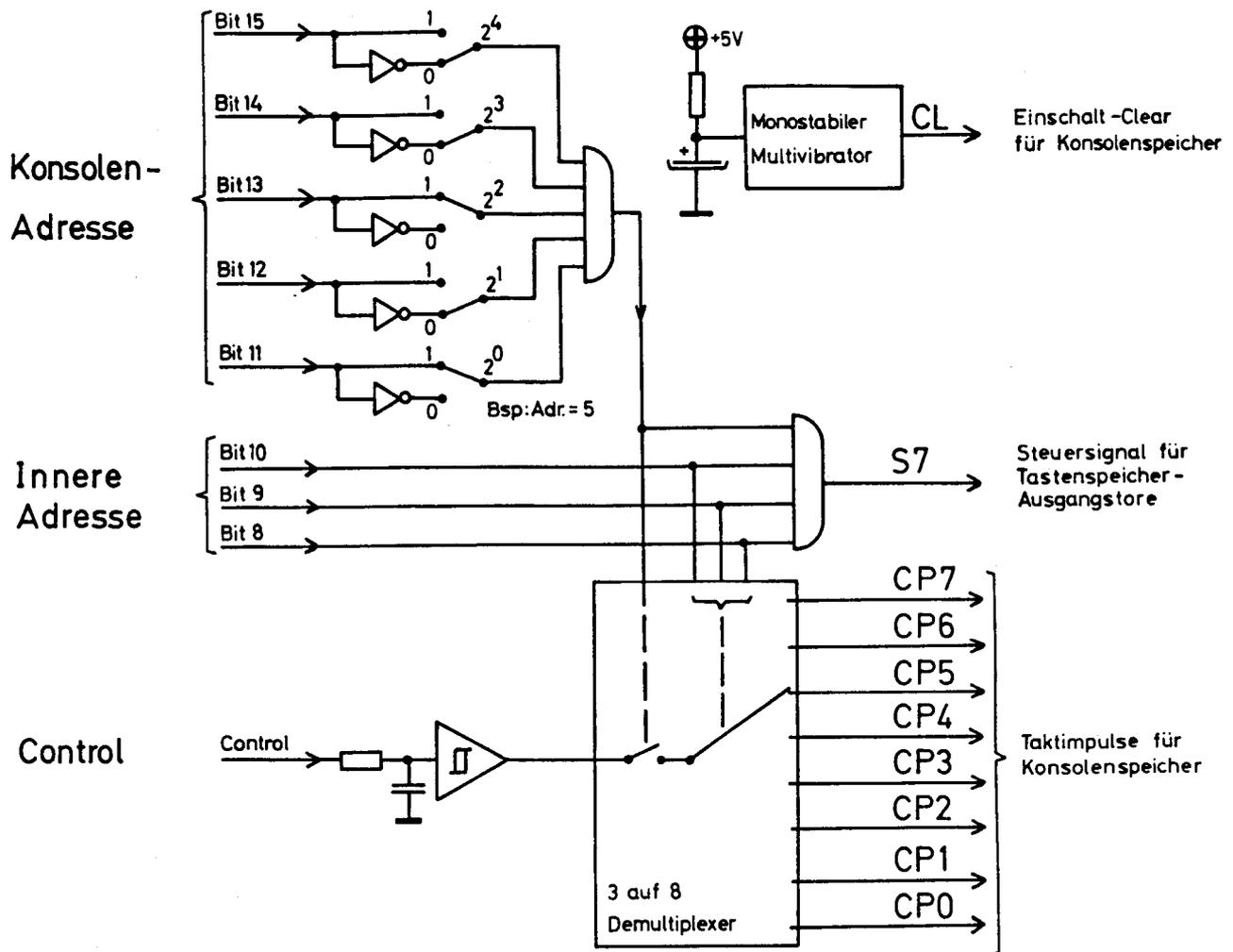


Fig 3.9 : Blockschema der Konsolensteuereinheit KS (Konsole 5)

Zur Detektion der Konsolenadresse werden zu den fünf Konsolenadressbits zunächst auch noch die inversen Signale gebildet. Je nach gewünschter Konsolenummer wird nun entweder das invertierte oder das nichtinvertierte Bit auf je einen Eingang eines UND-Tores mit 5 Eingängen geschaltet. Der Ausgang dieses Tores geht auf 1, wenn die Konsolenadresse stimmt. Dadurch wird eine Demultiplexerschaltung aktiviert, die das Systemtakt-Signal auf einen von acht Ausgängen schaltet, der durch die drei inneren Adressbits bestimmt wird. Zur Unterdrückung von eventuellen Uebersprechstörungen wird der Systemtakt vorher noch über ein Tiefpassfilter und einen Schmitt-Trigger geführt. Beim besonders kritischen Uebergang von 0 auf 1 werden dabei Störimpulse bis etwa $8\mu\text{s}$, beim viel harmloseren Wechsel von 1 auf 0 dagegen nur bis etwa $2\mu\text{s}$ Dauer unterdrückt.

Die Konsolensteuerung liefert daneben noch das wichtige Signal S7 für die Steuerung der Datenausgabe der Konsole. Sobald die richtige Konsolenadresse und die innere Adresse 7 (Tasteninput) vorhanden sind (also noch vor dem Erscheinen des Systemtakts), geht S7 auf 1 und öffnet die Ausgangstore des Tastenspeichers TA.

Ferner erzeugt die Konsolensteuerung kurz nach dem Erscheinen der Logik-Speisespannung den Einschalt-Clear-Impuls CL, der alle Konsolenspeicher auf 0 setzt, damit keine Fehlfunktionen unmittelbar nach dem Einschalten auftreten.

Das Taktsignal CP6 wird vorläufig nicht benützt, da die Konsole gegenwärtig nur über 7 Speicher verfügt. Es würde bei einem späteren Anschluss einer weiteren Konsolenperipherieeinheit als Takt für den dann nötigen 8. Speicher dienen.

3.6 Dianummern-Speicher und Projektorsteuerungen

Die Baugruppen DS und PS steuern die beiden Kodak S-RA-Anwahlprojektoren (siehe Fig. 3.10). Die gewünschte Dianummer für Projektor A wird in Speicher 0, diejenige für Projektor B in Speicher 1 gespeichert. Die beiden Speicher bestehen aus je 8 D-Flipflops und werden durch CP0 resp. CP1 getaktet.

Die beiden Projektoren melden ihre jeweilige Position, indem sie eine von neun "Zehner"-Leitungen mit einer von zehn "Einer"-Leitungen

Blockschema der Projektor - Steuerlogik

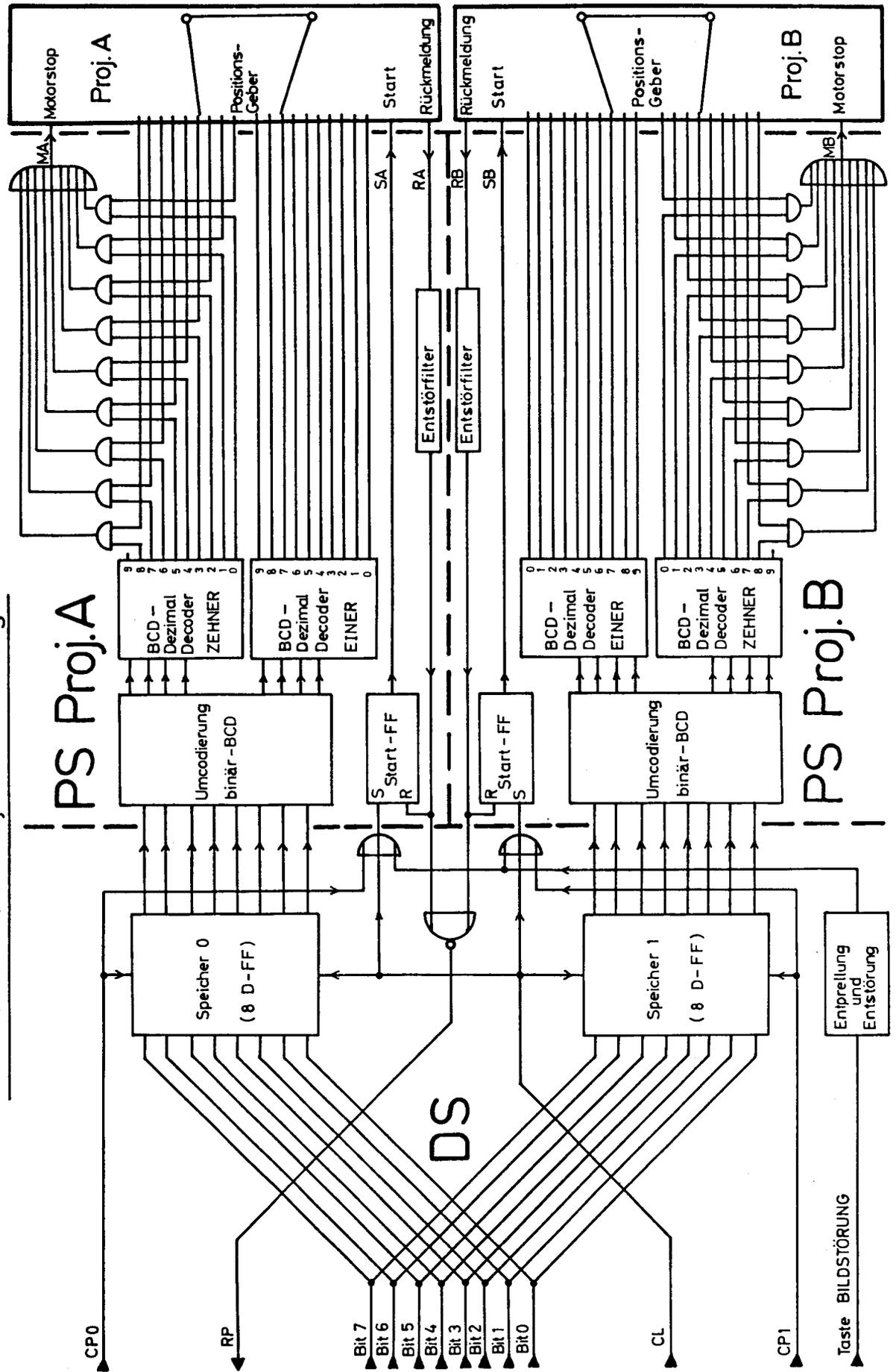


Fig. 3.10 : Blockschema der Projektor-Steuerlogik

verbinden. Wenn sich also z.B. ein Projektor auf Stellung 37 befindet, so ist die Zehnerleitung "30" über den Positionsgeber mit der Einerleitung "7" verbunden. Ein Bildwechsel wird durch einen speziellen Startimpuls (SA/SB) ausgelöst. Solange der Projektor sucht, gibt er ein Rückmeldesignal (RA/RB) aus. Sobald die richtige Position erreicht ist, muss der Transportmotor durch einen Stop-Impuls (MA/MB) angehalten werden. Darauf wird das neue Dia projiziert.

Die beiden Projektoren werden von je einer Projektorsteuerung PS gesteuert. Jeder Taktimpuls (CPO/CP1), der eine neue Dianummer in DS einliest, setzt ein RS-Flipflop, das einen Startimpuls (SA/SB) an den angeschlossenen Projektor abgibt und so den Diatransportvorgang einleitet. Dieses Flipflop wird beim Erscheinen der Such-Rückmeldung (RA/RB) des Projektors wieder gelöscht.

Die von DS kommende Dianummer (0 - 80, natürlicher Binärcode) wird zunächst in 2 BCD-Stellen umgewandelt. Jede dieser BCD-Stellen wird darauf einem BCD-Dezimal-Decoder zugeführt. Bei diesen Decodern wird immer nur der durch die BCD-Ziffer bestimmte Ausgang 1, die übrigen Ausgänge bleiben auf 0. Die Einer-Ausgänge werden nun auf die 10 Einer-Leitungen der Projektor-Positionsrückmeldung gegeben. Die Zehner-Ausgänge werden mit Hilfe von 9 UND-Toren mit den vom Projektor kommenden Zehnerleitungen multipliziert. Wenn während des Suchlaufs die richtige Einerstelle erreicht wird, schaltet der Projektor die vom Einer-Decoder stammende 1 auf die der aktuellen Position entsprechende Zehnerleitung. Nacheinander wird die 1 so auf die verschiedenen Zehnerleitungen geschaltet, bis auch die richtige Zehnerstelle erreicht ist. Das entsprechende UND-Tor geht dann auf 1 und liefert über das ODER-Tor, das die Ausgänge der 9 UND-Tore zusammenfasst, ein Motorstoppsignal (MA/MB) für den Projektor.

Neben der Speicherung der Dianummern erfüllt DS noch einige kleine Nebenaufgaben. Nach dem Einschalten der Konsole werden ja durch den Einschalt-Löschimpuls CL alle Konsolenspeicher gelöscht. Damit die Projektoren dann jedoch auch wirklich auf 0 laufen, muss CL auch einen Startimpuls auslösen. Bei Bildstörungen (z.B. wenn ein Dia nicht ganz in den Projektionsschacht fällt) soll durch eine Hilfstaste nochmals ein Suchvorgang ausgelöst werden können, ohne dass eine neue Dianummer eingelesen wird. Die beiden Rückmeldesignale RA und RB werden zu einem Signal RP (Ready Projektoren) zusammengefasst, das nur dann 1 ist, wenn beide Projektoren stillstehen.

3.7 Lampensteuerung (LS)

Die Baugruppe LS ist zunächst für die Ansteuerung der 5 eigentlichen Meldungslampen (RICHTIG, FALSCH, FEHLMANIPULATION, ZUM ASSISTENTEN und RESERVE) verantwortlich. Daneben sorgt sie für die Beleuchtung derjenigen Antworttasten, die jeweils gerade aktiv sind. Es gibt 5 mögliche Tastenbeleuchtungszustände: Alle Tasten (ausser Zusatzzeit), WEITER + STOP + 1 + 2 + 3 + 4 + 5, WEITER + STOP, START, keine Taste (dafür brennt WARTEN-Lampe). Die Beleuchtung der Taste ZUSATZZEIT ist unabhängig von den übrigen Tasten steuerbar. Schliesslich kann LS beim Eintreffen eines speziellen Befehls am Schluss eines Lehrprogramms das Abschalten der ganzen Konsole veranlassen.

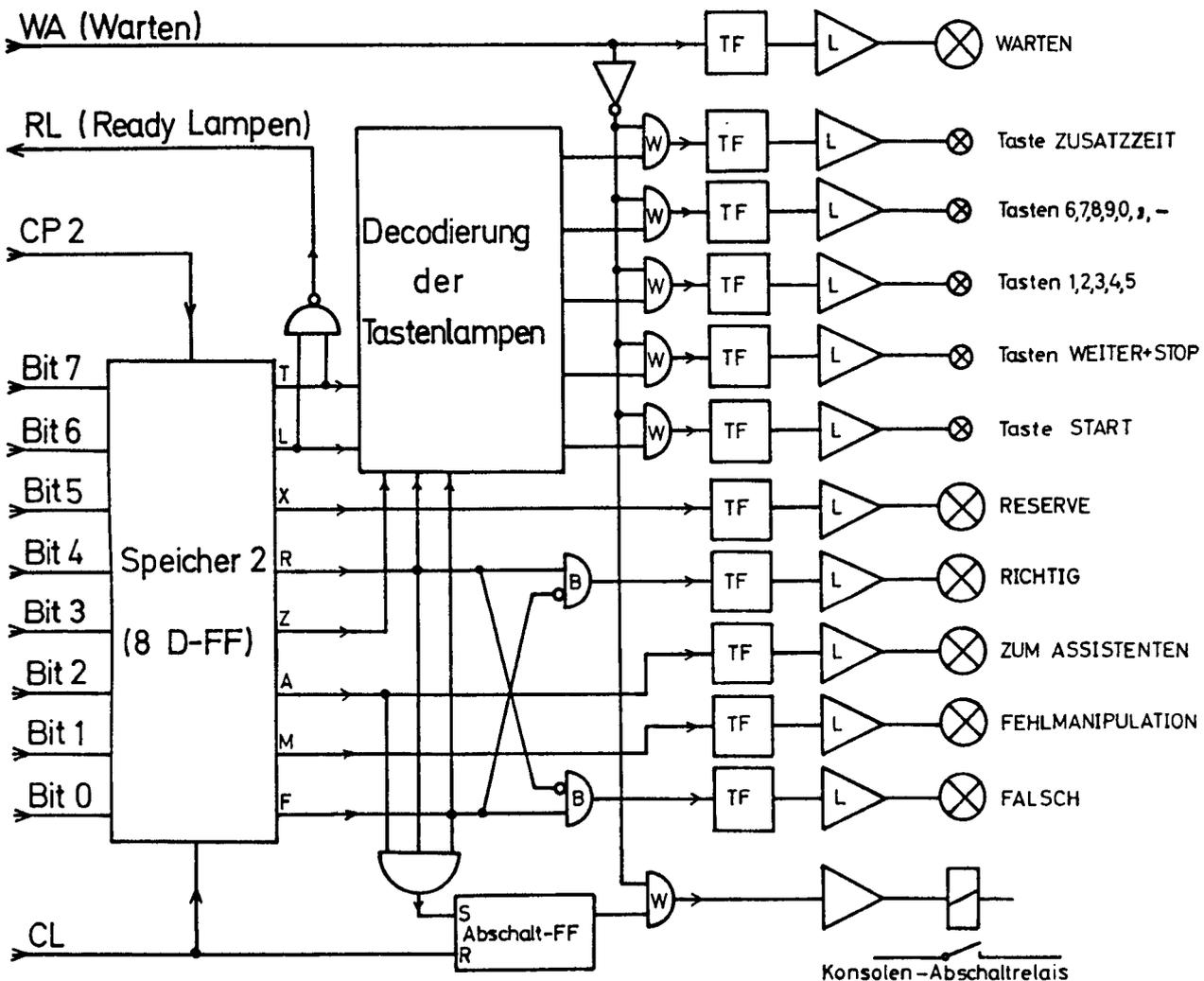
Für die Uebertragung dieser 12 Zustände stehen nur 8 Datenbits zur Verfügung. Die einzelnen Befehle müssen also mindestens teilweise codiert werden und können nicht völlig frei zusammengesetzt werden. Es ist sicher erwünscht, alle sinnvollen Kombinationen der Meldungslampen zuzulassen, damit möglichst viele verschiedene Gesamtmeldungen übertragen werden können. Für jede Meldungslampe wird somit im Prinzip 1 Bit benötigt. Alle Kombinationen, die gleichzeitig einen Ansteuerbefehl für die RICHTIG- und die FALSCH-Lampe enthalten, sind jedoch sinnlos und können für die Uebertragung weiterer Befehle verwendet werden.

Die Lampenbefehle werden folgendermassen codiert :

		<u>Meldungslampen, von System und Lehrprogramm steuerbar:</u>								
		X = 1 : RESERVE								
		R = 1 : RICHTIG								
		A = 1 : ZUM ASSISTENTEN								
Bit-Nr. :		M = 1 : FEHLMANIPULATION								
7 6 5 4 3 2 1 0		F = 1 : FALSCH								
<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>T</td><td>L</td><td>X</td><td>R</td><td>Z</td><td>A</td><td>M</td><td>F</td></tr></table>	T	L	X	R	Z	A	M	F		<u>Tastenlampen, nur vom System steuerbar:</u>
T	L	X	R	Z	A	M	F			
Datenbits		Z = 1 : Keine ZUSATZZEIT								
(für Speicher 2)		TL = 00 : Alle Tasten (ausser ZUSATZZEIT)								
		TL = 01 : WEITER, STOP, 1, 2, 3, 4, 5								
		TL = 10 : WEITER, STOP								
		TL = 11 : Tasten unbeleuchtet, WARTEN brennt								
		R=1,F=1 : START								
		<u>Sonderfunktion:</u>								
		R=1,A=1,F=1 : Konsole schaltet ab								

Fig. 3.11 : Die Codierung der Lampenbefehle

Fig. 3.12 zeigt den Aufbau der Lampensteuerung. Alle Lampen werden aus einer separaten 24V-Spannungsquelle gespeist. Dadurch können die Lampenströme klein gehalten werden (20mA für Tastenlampen, 50mA für Meldungslampen). Zur weiteren Verringerung von Störungen wegen zu grossen di/dt (besonders beim Einschalten der Lampen) werden die Massen der 5V-Logikspeisung und der Lampenspeisung getrennt zugeführt. Sie sind nur über einen niederohmigen Widerstand miteinander verbunden. In jeder Lampen-Steuerleitung befindet sich überdies ein Tiefpassfilter TF. Dieses verlangsamt das Ein- und Ausschalten der Lampe (deshalb kleineres di/dt) und dämpft Rückwirkungen von den Lampentreiberschaltungen L auf die TTL-Steuerlogik.



B = Blockiertore, verhindern gleichzeitige RICHTIG-/FALSCH-Anzeige

L = Lampentreiberschaltungen

TF= Tiefpassfilter

W = Sperrtore, verhindern Tastenlampenbeleuchtung bei gesperrter Konsole

Fig. 3.12 : Blockscheema der Lampensteuerung

Die ankommenden Lampenbefehle werden mit Hilfe von CP2 in den Speicher 2 eingelesen. Die Signale für die Meldungslampen RESERVE, ZUM ASSISTENTEN und FEHLMANIPULATION gelangen direkt von den Speicherausgängen zu den Lampentreibern. Dagegen müssen bei RICHTIG und FALSCH die beiden UND-Tore B das Anzünden der Lampen verhindern, wenn sowohl $R = 1$ als auch $F = 1$ ist, da diese Kombination für andere Befehle (START, ABSCHALTEN) ausgenutzt wird.

Die Steuersignale für die Tastenlampen müssen aus den Signalen T, L, Z, R und F gewonnen werden. Dies besorgt die Tastenlampen-Decodierung. Wenn $TL = 11$ ist, wird das Signal $RL = 0$ und sperrt den Tastenspeicher TA. Wenn dieser aus irgend einem Grund gesperrt ist, wird ein Warten-Signal $WA = 1$ ausgegeben, das in LS die WARTEN-Lampe anzündet. $WA = 1$ blockiert aber auch die Weiterleitung der decodierten Signale durch die Sperrtore W und verhindert somit das Anzünden der Tastenlampen. Wenn R, A und F alle gleichzeitig 1 sind, wird über ein 3-Eingangs-UND-Tor das Abschalt-Flipflop gesetzt. Sobald $WA = 0$ ist, wird darauf das Abschaltrelais angesteuert, das den Haltekreis des Konsolen-Hauptrelais unterbricht und so die Stromzufuhr der Konsole abschaltet.

3.8 Zeitsteuerung und Zeitanzeige (ZS)

Die Präsentationszeit der einzelnen Bilder kann durch das Lehrprogramm begrenzt werden, damit die Lehrkonsolen nicht durch einzelne Studenten übermäßig lange besetzt bleiben oder sogar längere Zeit unbenutzt weiterlaufen. Die noch verbleibende Antwortzeit (Restzeit) wird durch ein Leuchtdiodenband angezeigt. Je mehr Leuchtdioden brennen, desto mehr Zeit ist noch vorhanden. Die Zeiten von 0 bis 60 Sekunden werden zur Erzielung einer möglichst feinen Auflösung in 10-Sekunden-Stufen angezeigt (eine Leuchtdiode = 10 Sekunden). Für Zeiten über 60 Sekunden genügt ein gröberes Raster mit Stufen von $\frac{1}{2}$ bis 1 Minute (zwei etwas hellere Leuchtdioden = 1 Minute). Zeiten über 5 Minuten zünden alle Leuchtdioden der Zeitanzeige an.

Fig. 3.13 zeigt das Blockschema der Zeitsteuerung. Die anzuzeigende Restzeit wird mit Hilfe von CP3 in 10-Sekunden-Einheiten in den Speicher 3 eingelesen. Zeiten von 320s und mehr steuern über ein 3-Eingangs-ODER-Tor direkt die $>5\text{Min}$ -Leuchtdioden an. Zeiten von 310s und weniger werden zuerst einem 5 auf 32 Demultiplexer zugeführt. Nur einer der 32 Ausgänge wird jeweils logisch 1 und aktiviert die angeschlossene Decoderschaltung, die über eine Treiberschaltung die entsprechende(n) Leuchtdiode(n) anzündet. Wenn

eine Leuchtdiode mit einem bestimmten Zeitgewicht angesteuert wird, zündet die Steuerlogik automatisch auch alle Leuchtdioden mit geringerem Gewicht an. Falls die Restzeit auf 10s oder weniger gesunken ist, wird ein Signal ZA (Zeitablauf) erzeugt. Dadurch wird einerseits direkt ein akustisches Signal (Gongschlag) produziert und andererseits die Baugruppe EV veranlasst, eine akustische Warnung in den Tonkanal einzublenden. Der Adressat kann dann entweder durch Drücken der Taste ZUSATZZEIT mehr Zeit verlangen oder die Frage noch beantworten, bevor die Restzeit 0 wird und die Konsole automatisch zum nächsten Bild weiterschaltet.

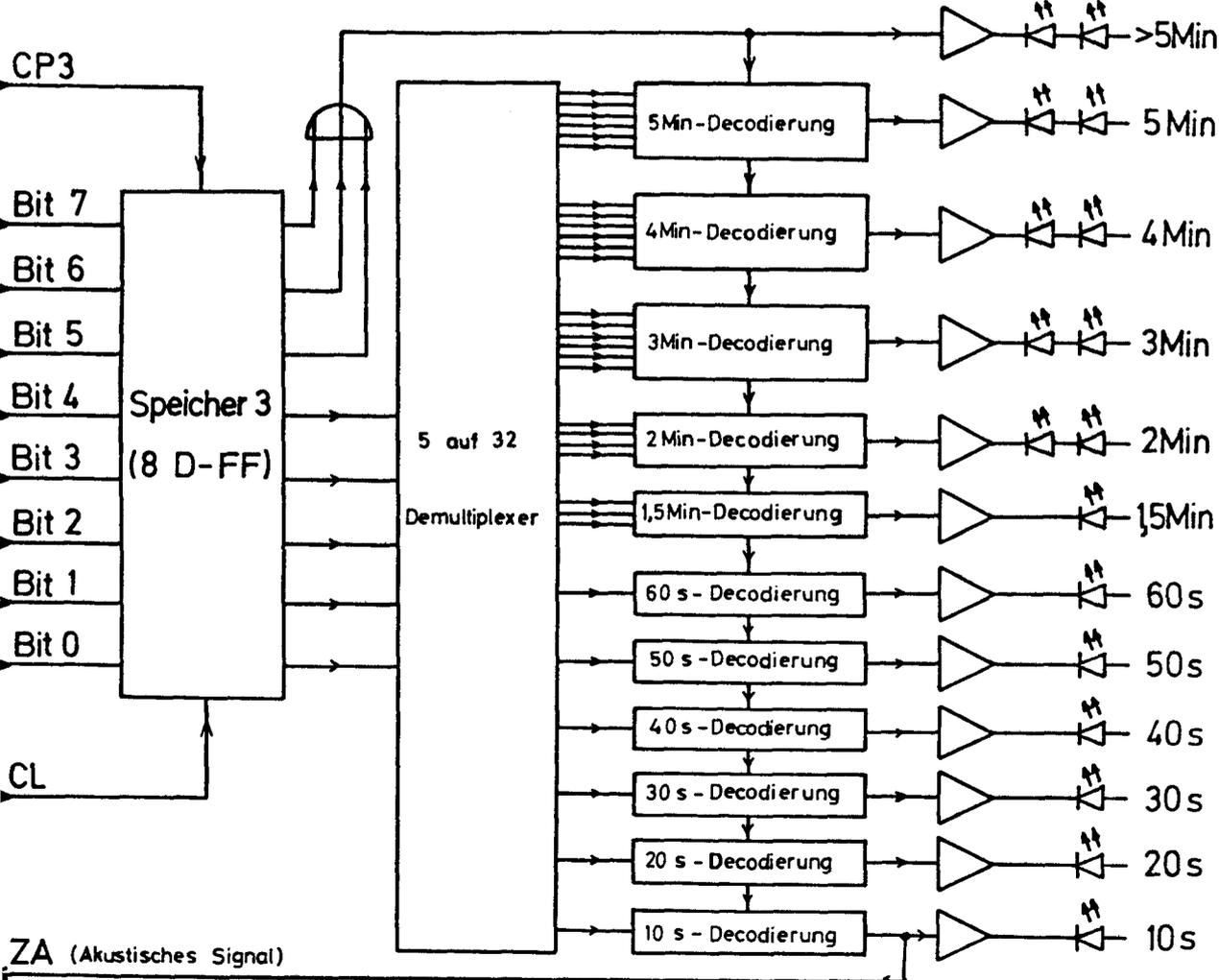


Fig. 3.13 : Blockscheema der Zeitsteuerung und Zeitanzeige (ZS)

3.9 Numerische Anzeige (NA)

Die Baugruppe NA ist in die Frontplatte der Lehrkonsole eingebaut. Sie erlaubt die Anzeige von Zahlen mit maximal 8 Stellen. Jede der verwendeten Ziffernanzeigen besitzt bereits einen eingebauten Speicher und kann ausser den Zahlen 0 - 9 auch das Minuszeichen sowie nichts (BLANK) anzeigen. Davon unabhängig kann bei jeder Ziffer der Dezimalpunkt angesteuert werden. Für die Adressierung der 8 Ziffern werden 3, für den Dezimalpunkt 1 und für den Zahlenwert 4 Datenbits benötigt. Pro Datenausgabezyklus kann also nur eine Ziffer übertragen werden. Für eine vollständige 8-stellige Zahl sind somit 8 unmittelbar aufeinanderfolgende Datenübertragungszyklen notwendig. Das Format der angezeigten Zahlen ist von der Hardware her nicht eingeschränkt. Das gegenwärtige PLANETH-Systemprogramm kann 8- und 4-stellige ganze Zahlen (für ETH-Nummer und Lehrprogrammnummer), ein 4-stelliges Spezialformat für die laufende Anzeige der erzielten Punktzahl sowie reelle Zahlen anzeigen (4 Stellen Mantisse, 2 Stellen Exponent, je 1 Stelle Vorzeichen, Beispiel siehe Fig. 3.14) .

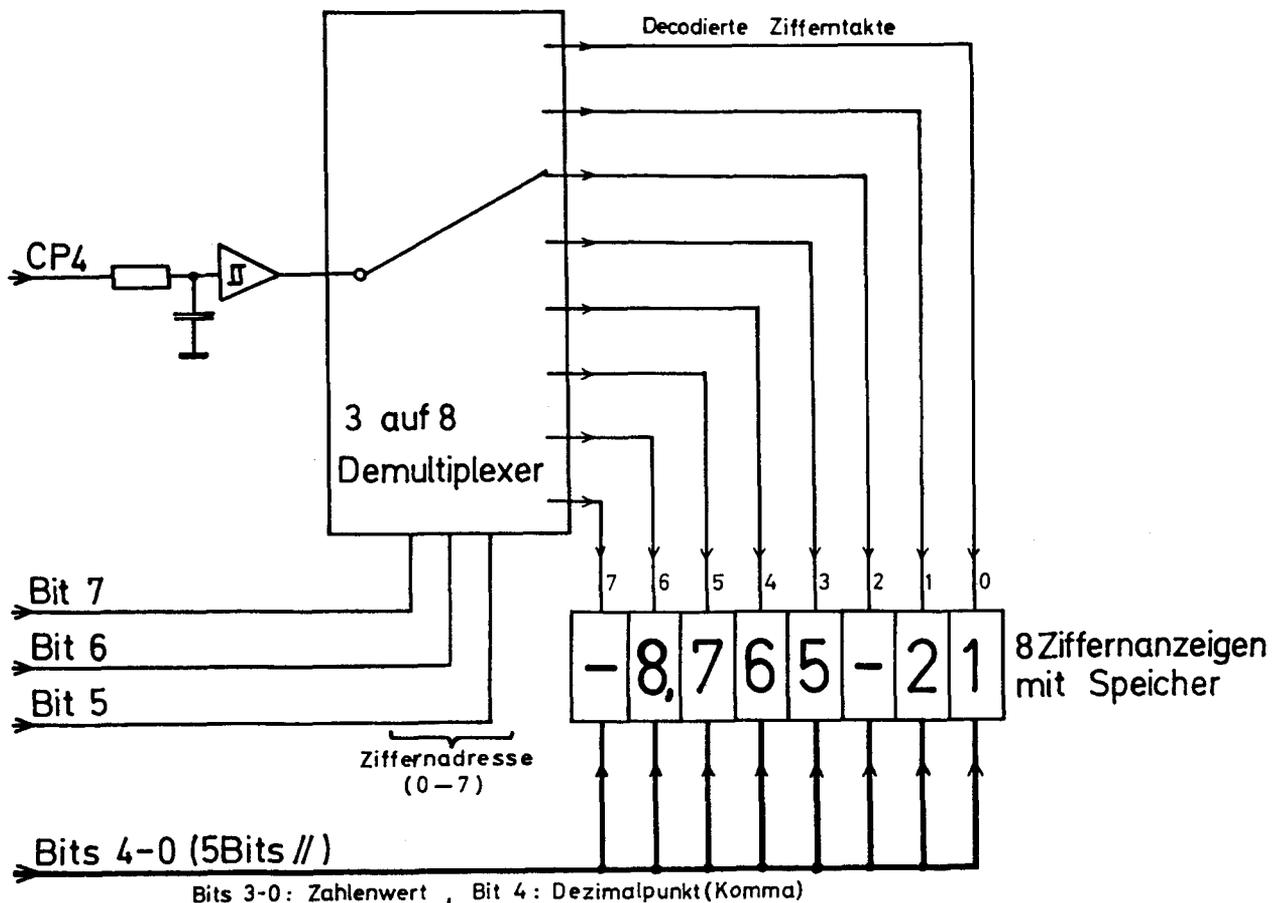


Fig. 3.14 : Blockscheema der numerischen Anzeige (NA)

Dank den eingebauten Ziffernspeichern wird die Schaltung der Baugruppe NA sehr einfach (siehe Fig. 3.14). Zur Unterdrückung möglicher Störungen auf der Verbindungsleitung zwischen der Konsolenelektronik und NA wird CP4 zunächst über ein Tiefpassfilter und anschliessend auf einen Schmitt-Trigger geführt. Aus diesem Signal werden mit einem 3 auf 8 Demultiplexer, der von den Ziffernadressbits 7 - 5 gesteuert wird, die Takte für die einzelnen Ziffernspeicher gewonnen. Jede Ziffernanzeige ist ferner an die Datenbits 4 - 0 angeschlossen. Sie übernimmt die für sie bestimmten Daten, wenn ihr Zifferntakt 1 wird und zeigt den gespeicherten Zahlenwert laufend an.

3.10 Das PLANETH-Kassetten-Tonbandgerät

Schon bald nach den ersten Versuchen mit computergesteuerten Lehrkonsolen, die ihre Informationen nur mit Diaprojektoren präsentierten, tauchte der Wunsch nach einem geeigneten Kassettentonbandgerät auf, um die Anlage zu einem vollwertigen audiovisuellen Lehrsystem auszubauen. Das Gerät musste voll fernsteuerbar und so konzipiert sein, dass mit einer geeigneten Steuerelektronik ein beliebiger Zugriff zu den einzelnen Sprach- und Tonsequenzen möglich war, d.h. es musste unbedingt auch im schnellen Vor- und Rücklauf Information ab Band gelesen werden können. Da jedoch kein passendes Gerät auf dem Markt war, musste es unter Verwendung der 3-motorigen Kassettenmechanik URTC-3M der Firma Lenco selbst entwickelt werden.

Bei der gewählten Kassettenmechanik sind alle Funktionen elektrisch steuerbar. Es sind zwei Gleichstrom-Wickelmotoren und ein Wechselstrom-Synchronmotor für den Tonwellenantrieb vorhanden. Dieser Motor läuft dauernd. Bei Aufnahme und Wiedergabe wird die Andruckrolle von einem Andruckmagneten gegen die Tonwelle gepresst. Beim schnellen Vor- und Rücklauf schiebt ein Hubmagnet die Tonkassette etwas zurück, so dass das Band ungefähr um 0,2mm von den Köpfen abgehoben wird. Dieser Abstand ist gross genug, dass keine Verschleisserscheinungen auftreten. Er ist aber noch so klein, dass sehr tiefe Aufnahmefrequenzen (<30Hz) auch im Suchlauf noch einwandfrei lesbar sind, wobei allerdings die Wiedergabefrequenz um den Faktor der Geschwindigkeitserhöhung höher liegt (siehe Kap. 3.11.1). Ein weiterer Magnet erlaubt das Auswerfen der eingelegten Kassette.

Fig. 3.15 zeigt den prinzipiellen Aufbau der Tonband-Elektronik.

Zur Erhöhung der Störsicherheit wurde sie in CMOS-Logik mit 12V Speisepannung realisiert.

Alle ankommenden Steuerimpulse werden zunächst mit Tiefpassfiltern und Schmitt-Triggern entstört. Die Signale VOR, RUECK, PLAY (Wiedergabe) und AUFN (Aufnahme) gelangen anschliessend über die Eingangstore in den Befehlsspeicher. Sobald ein Befehl im Speicher vorhanden ist, geht BUSY auf 1 und verhindert das Einlesen weiterer Steuersignale.

Der Hubmagnet ist normalerweise angezogen, das Band etwas von den Köpfen abgehoben und das Hub-Rückmeldesignal HR ist deshalb auf 1. Wenn jedoch ein PLAY- oder AUFN-Signal vorliegt, wird er von der Hubmagnet-Steuerung abgeschaltet. Nach dem Einschalten des Gerätes oder nach dem Einschieben einer neuen Kassette sorgt die Hubmagnet-Steuerung dafür, dass der Magnet etwas verzögert den Anzugbefehl bekommt, damit sich der Hubmagnet-Kondensator zuerst aufladen kann.

Ein PLAY- oder AUFN-Befehl bewirkt also zunächst das Abfallen des Hubmagnets. Erst wenn die Hub-Rückmeldung HR 0 geworden ist, d.h. wenn das Band wieder auf den Köpfen aufliegt, kann der Befehl die Langsamlauf-Ausgangstore passieren und auf die entsprechende Rückmeldeleitung (PLAY-R resp. AUFN-R) für die Steuerelektronik gelangen. Das Signal AM wird ebenfalls 1 und sorgt für das Anziehen des Andruckmagneten, den Bandtransport und den richtigen Bandzug. Bei Aufnahme wird zusätzlich der 50kHz-Löschgenerator eingeschaltet und die Aufnahme-/Wiedergabeköpfe auf die Ausgänge der Aufnahmeverstärker geschaltet.

Wenn hingegen ein VOR- oder RUECK-Befehl vorliegt, bleibt der Hubmagnet angezogen und HR somit auf 1. Der Befehl kann deshalb die Suchlauf-Ausgangstore passieren und auf die entsprechenden Rückmeldeleitungen (VOR-R resp. RUECK-R) gelangen. Die angeschlossenen Motorschalttransistoren legen bei VOR den linken, bei RUECK den rechten Wickelmotor an die volle Motorspeisepannung.

Für das Löschen eines Befehls gibt es drei Möglichkeiten :

- a) Einen STOP-Impuls auf den STOP-Eingang.
- b) Ein Einschalt-Clear-Signal unmittelbar nach dem Einschalten des Gerätes.
- c) Einen BANDENDE-STOP, wenn BUSY = 1 ist und wenn der Phototransistor PT leitend wird, weil das durchsichtige Kassettenvorspannband das Licht der Bandende-Leuchtdiode durchlässt. Damit man auch wieder von dieser Bandstelle wegkommen kann, muss BUSY etwas verzögert werden.

Sobald ein Befehl gelöscht wird, geht BUSY auf 0. Die Eingangstore werden aber erst ca. 250ms später wieder freigegeben, damit das Gerät erst wieder

neue Befehle akzeptiert, wenn das Band sicher stillsteht. Erst dann wird auch die STOP-Rückmeldung auf die Steuerelektronik ausgegeben. Wenn BUSY = 0 und eine Kassette eingeschoben ist ($KE = 1$), wird $BS = 1$ und bewirkt, dass beide Wickelmotoren ein kleines gegensinniges Drehmoment erhalten, damit das Band auch im Stillstand immer schön gestrafft bleibt.

Der Kassetten-Auswurfmagnet kann nur betätigt werden, wenn das Signal AUSWURF-ERLAUBNIS 1 ist, das Tonbandgerät stillsteht und wenn die Auswurf Taste gedrückt wird.

Die Schaltung der beiden Wickelmotoren enthält noch einige Besonderheiten. Beim schnellen Vorlauf schaltet der Transistor T_1 voll durch, d.h. am linken Motor liegt die volle Motorspannung. Der Ankerstrom dieses Motors fließt durch den Widerstand R_A . Der rechte Motor arbeitet als Generator. Die von ihm erzeugte Spannung liegt über die Diode D_2 am Suchlauf-Geschwindigkeitsregler. Je langsamer der Antriebsmotor läuft, je mehr Strom er also aufnimmt, desto weniger wird die Generatorspannung des Abwickelmotors vom Regler belastet und desto geringer ist die Bremswirkung. Die Bandgeschwindigkeit im Suchlauf kann so mit einfachen Mitteln bis auf etwa +/- 10% Abweichung vom Sollwert konstant gehalten werden, was die zuverlässige Detektion der Sequenzmarkierungen sehr erleichtert (siehe Kap. 3.11).

Wenn der Antriebsmotor wieder abgeschaltet wird, erhält die Basis von T_4 über die RC-Serieschaltung einen kurzen Stromstoß. Der Transistor T_4 wird kurzzeitig leitend und legt den Abwickelmotor an eine Bremspannung. Zunächst liegt die volle Motorspeisespannung über dem Abwickelmotor, allmählich lädt sich aber der in der Emitterleitung liegende Kondensator auf, wodurch die Bremspannung um die Zenerdiodenspannung verringert wird. Auf diese Weise kann ein sehr rascher Bandstop bei minimalem Gegenlauf des Bandes erreicht werden.

Beim schnellen Rücklauf funktioniert alles analog, es wird jedoch der rechte Motor angetrieben und der linke Motor gebremst.

Die gesamte Tonband-Elektronik ist auf vier Prints verteilt. Die Aufnahme-/Wiedergabeverstärker sind in einem vollständig abgeschirmten Gehäuseabteil untergebracht. Auf den übrigen drei Prints befindet sich die Elektronik, die die Magnete und Motoren steuert.

Blockschema des PLANETH-Kassetten - Tonbandgerätes

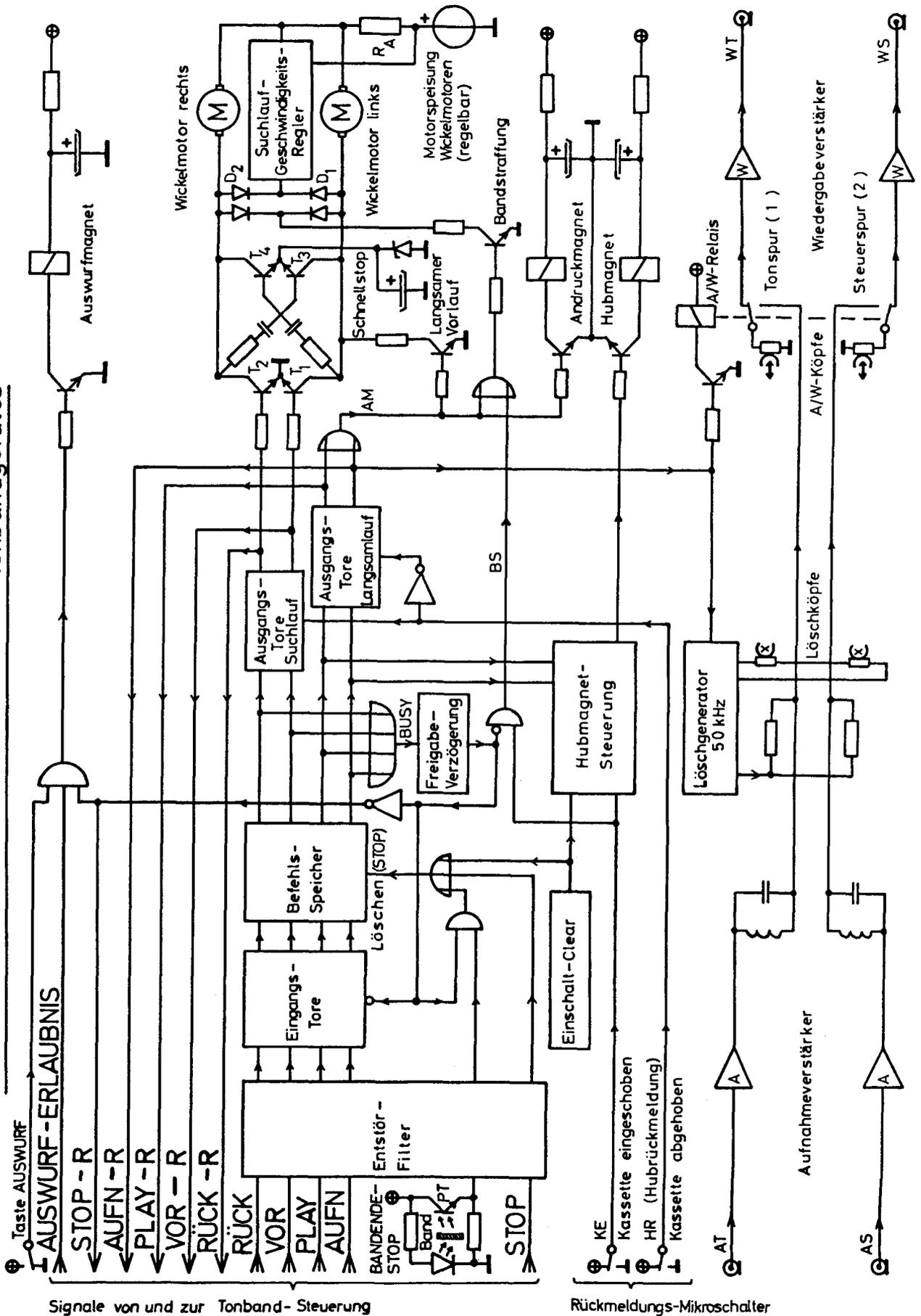


Fig. 3.15 : Blockschema des PLANETH-Kassetten-Tonbandgerätes

3.10.1 Wichtigste technische Daten des PLANETH-Kassettenbandgeräts

Wenn nicht anders angegeben : Typische Werte

Spuren	: 2
Kanäle	: 2 (Stereo)
Bandgeschwindigkeit	: 4,75 cm/s
Geschwindigkeit im Suchlauf	: ca. 45-fache Normalgeschwindigkeit
Bremsweg aus Suchlauf	: <17 cm (entspricht <3,5s Spielzeit)
Frequenzgang (DIN 45511)	: 100 Hz - 5 kHz
Geräuschabstand	: >40 dB
Uebersprechdämpfung	: >34 dB
Bandart	: Fe ₂ O ₃ , 3,81 mm breit
Eingangsspannung	: 7 mV
Eingangsimpedanz	: 30k Ω //1nF
Ausgangsspannung	: 400 mV
Ausgangsimpedanz	: 500 Ω

Alle Funktionen sind elektrisch fernsteuerbar.

Für weitere Daten, Detailschemas und ausführlichere Funktionsbeschreibungen sei auf [2] verwiesen.

3.11 Tonbandsteuerung

3.11.1 Das Konzept der Tonsequenzmarkierung

Das PLANETH-Tonbandgerät hat die Aufgabe, Sprach- und Tonsequenzen, also analoge Signale, wiederzugeben. Diese werden auf einer der beiden Spuren, der Tonspur, aufgenommen. Für die notwendigen Markierungen der einzelnen Sequenzen werden deshalb zweckmässigerweise auch analoge Signale verwendet. Sie werden auf der andern Spur, der Steuerspur, aufgezeichnet.

Eine sehr einfache Möglichkeit ist die Verwendung einer Anzahl von Sinusschwingungen einer bestimmten Frequenz, die ein tonfrequenten Impulspaket oder kurz eine "NF-Marke" bilden. Der Anfang jeder Sequenz wird mit einem solchen Impulspaket markiert. Die Steuerlogik zählt diese NF-Marken laufend und kennt so immer die aktuelle Position

des Bandes. Bei Verwendung dieser Konzeption wirkt sich jedoch bereits ein einziger Störimpuls verheerend aus, denn er bewirkt einen bleibenden, nicht mehr automatisch korrigierbaren Fehler. Beim Entwurf und Aufbau einer solchen Steuerlogik muss deshalb die grösstmögliche Störsicherheit angestrebt werden. Natürlich wurden auch andere Markierungskonzepte untersucht. Es zeigte sich aber, dass diese entweder eine viel zu lange Markendauer zur Folge hatten oder dann doch nicht alle möglichen Störfälle korrigieren konnten.

Um einen beliebigen Zugriff (random access) zu den einzelnen Sequenzen zu ermöglichen, müssen die Markierimpulse auch im schnellen Suchlauf gelesen werden können. Wenn das Band dabei jedoch den Tonkopf berührt, tritt wegen der hohen Suchgeschwindigkeit sowohl am Band als auch am Tonkopf rasch ein sehr grosser Verschleiss auf. Das Band muss deshalb im Suchlauf etwas vom Tonkopf abgehoben werden. Dies beeinträchtigt jedoch die Wiedergabe der mittleren und hohen Frequenzen sehr stark, denn nach [3] gilt für die induzierte Hörkopfspannung bei tiefen Frequenzen folgende Näherungsformel :

$$U_H = \phi_K \cdot V_S \cdot \omega_A \cdot e^{-2\pi \frac{d}{\lambda}} = \phi_K \cdot V_S \cdot \omega_A \cdot e^{-d \frac{\omega_A}{v_0}} \quad (3.5)$$

wobei U_H = induzierte Hörkopfspannung

ϕ_K = Kurzschlussfluss

ω_A = Aufnahme Frequenz

v_0 = Bandgeschwindigkeit bei Aufnahme und Wiedergabe (4,75cm/s)

$\lambda = v_0 / f_A = 2\pi v_0 / \omega_A$ = Aufgezeichnete Wellenlänge

$V_S = v_{\text{Suchlauf}} / v_0$ = Normierte Suchgeschwindigkeit

d = Abstand Band-Tonkopf

Man sieht, dass ziemlich tiefe Aufnahme Frequenzen verwendet werden müssen, damit die Hörkopfspannung auch bei einem Abstand von einigen 1/10 mm nicht allzu stark abfällt. Andererseits kann die Aufnahme Frequenz nicht allzu tief gewählt werden, da sonst vor allem beim Kopieren solcher Aufnahmen infolge der unteren Bandgrenze der verwendeten Tonbandgeräte Probleme auftreten. Nach vielen Versuchen wurde für die niederfrequente Sequenzmarkierung ein 25Hz-Sinussignal von $T_A = 1,5s$ Aufnahmedauer gewählt.

Im schnellen Suchlauf verändern sich natürlich Frequenz und Dauer der Marken in Funktion der normierten Suchgeschwindigkeit V_S .

Für die Frequenz ω_s im Suchlauf gilt :

$$\omega_s = V_s \cdot \omega_A \quad (3.6)$$

und für die Dauer T_s einer Marke im Suchlauf :

$$T_s = T_A / V_s \quad (3.7)$$

Diese Veränderungen müssen bei der Dimensionierung der Markendetektionsschaltung (in Baugruppe MD) berücksichtigt werden.

Zur möglichst weitgehenden Unterdrückung von Störungen ist es angezeigt, den Markendetektor nur dann zu aktivieren, wenn er wirklich benötigt wird, also im Suchlauf. Die Steuerelektronik ist so auszuliegen, dass in dieser Zeit keine Magnete geschaltet werden müssen. Alle vom Tonband auf die Steuerelektronik kommenden logischen Signale sind mit Tiefpassfiltern und Schmitt-Triggern zu entstören, und die Speisespannung für die Tonbandelektronik soll aus einer separaten Quelle direkt zum Tonbandgerät geführt werden.

Wegen der grossen Geschwindigkeit im Suchlauf kann das Gerät beim Auffinden der richtigen Marke natürlich nicht augenblicklich anhalten, sondern es wird ein gewisser Bremsweg benötigt. Da die Marken nur im Suchlauf gelesen werden sollen, wird auch eine gewisse Beschleunigungsstrecke benötigt, bis nach einem Halt V_s wieder so gross geworden ist, dass die Frequenz ω_s und die Dauer T_s eine sichere Detektion gestatten. Brems- und Beschleunigungsweg wachsen etwa proportional zur normierten Suchgeschwindigkeit V_s . Je grösser V_s ist, desto grösser wird also auch der nötige Sicherheitsabstand zwischen Markende und Tonsequenzanfang einerseits und Tonsequenzende und Marken-anfang andererseits, damit in jedem Fall nur die richtige Sequenz, diese aber vollständig, abgespielt wird (vergl. Fig. 3.16). Dagegen hat ein grosses V_s den Vorteil, dass die Zugriffszeit zu den einzelnen Sequenzen klein ist, was natürlich sehr erwünscht ist. Auch hier war wieder ein Kompromiss erforderlich : Es wurde $V_s = 45$ gewählt.

Damit immer eine ausreichende Beschleunigungsstrecke vor der nächsten NF-Marke zur Verfügung steht, muss das Tonbandgerät nach dem Abspielen der gewünschten Sequenz durch eine zweite Marke gestoppt werden, die im Suchlauf sicher nicht gelesen werden kann. Die Frequenz

dieser Marke soll deshalb möglichst hoch sein (vergl. Formel (3.5)), jedoch noch deutlich unter der oberen Grenzfrequenz des Tonbandgerätes liegen. Für diese "HF-Marke" wurde ein 3,2kHz-Sinussignal von $\frac{1}{2}$ s Dauer gewählt.

Gemäss Kap. 3.10.1 beträgt der Bremsweg bei $V_S = 45$ höchstens 17cm (entsprechend 3,5s Spielzeit). Wenn auf jeder Seite noch $\frac{1}{4}$ s Sicherheitsabstand vorgesehen werden, sieht also eine vollständige Sequenzmarkierung in Funktion der Zeit wie folgt aus :

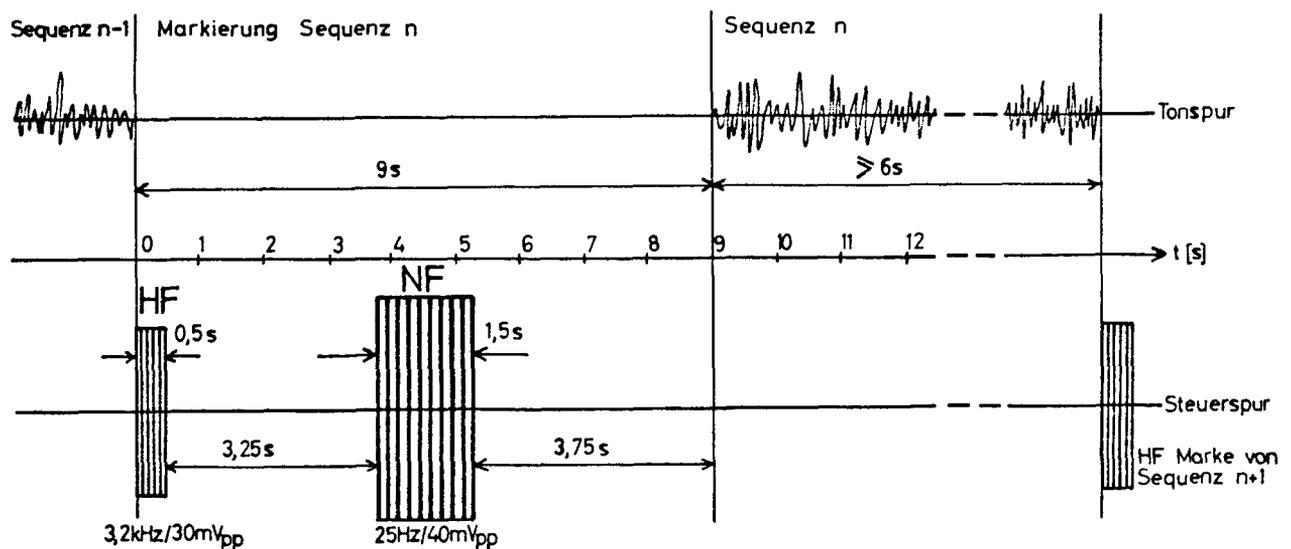


Fig. 3.16: Eine Sequenzmarkierung in Funktion der Zeit (bei Aufnahme)

Diese Sequenzmarkierungen werden bei der Aufnahme mit einem einfachen Hilfsgerät erzeugt, das auch dafür sorgt, dass die einzelnen Tonsequenzen eine minimale Länge von 6 Sekunden haben. Dank dem so erzielten minimalen Abstand von 13,5s zwischen zwei NF-Marken können in der Steuerelektronik weitere Schutzmassnahmen gegen Störimpulse vorgesehen werden (siehe Kap. 3.11.3) .

Die für die Sequenzmarkierungen benötigte Zeit steht natürlich für die eigentlichen Tonaufnahmen nicht mehr zur Verfügung. Die für die Aufnahme der Tonsequenzen noch verbleibende Zeit berechnet sich wie folgt :

$$t_T = t_K - n_S \cdot 9s \quad (3.8)$$

wobei t_T = Für Tonaufnahmen noch verfügbare Gesamtzeit

t_K = (Einseitige) Spieldauer der verwendeten Kassette
(1800s bei C60-er, 2700s bei C90-er Kassetten)

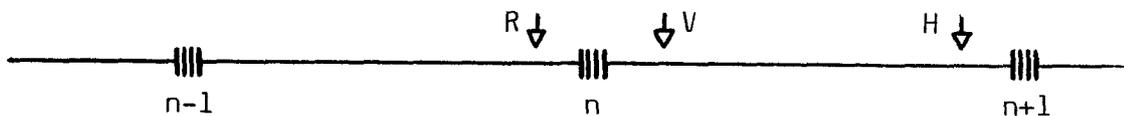
n_S = Gesamte Anzahl Tonsequenzen (≤ 127)

Bei der gewählten normierten Suchgeschwindigkeit $V_s = 45$ dauert das Umspulen der für unsere Anwendungen gut geeigneten C60-Kassetten immerhin noch 40s. Es ist deshalb zweckmässig, für das Uberspringen grösserer Strecken auf dem Band die Möglichkeit der Vorpositionierung zu schaffen. Wenn das Tonbandgerät gerade nichts abspielen muss, kann es an den Anfang oder mindestens in die Nähe der nächsten abzuspielenden Sequenz laufen. Erhält es dann später den Befehl, diese Sequenz abzuspielen, so ist die Zugriffszeit wesentlich kleiner und meist vernachlässigbar. Für die Unterscheidung zwischen Positionier- und Abspielbefehlen wird eines der acht Datenbits benötigt, nämlich das Bit 7 (Bit 7 = 1 : Abspielbefehl, Bit 7 = 0 : Positionierbefehl). Mit den verbleibenden 7 Bits (Bits 6 - 0) können dann maximal 127 verschiedene Sequenzen adressiert werden.

3.11.2 Der Ablauf der Suchvorgänge beim gewählten Markierungskonzept

Die Sequenzmarkierungen befinden sich immer vor dem Anfang der zugehörigen Tonsequenz. Anhand von Fig. 3.17 sollen nun die Auswirkungen mehrerer aufeinanderfolgender Such- und Abspielvorgänge untersucht werden.

Wenn die Sequenz n im Vorwärtslauf aufgesucht worden ist, so stoppt der Tonkopf in Stellung V, also rechts von der Marke n. Wird auch noch das Abspielen dieser Sequenz verlangt, so gelangt der Tonkopf bis in Stellung H (HF-Stopmarke vor NF-Marke n+1), er befindet sich also immer noch rechts von der Marke n. Wird die Sequenz n jedoch im Rückwärtslauf aufgesucht, so hält der Tonkopf in Stellung R links von der Marke n. Bei anschliessendem Abspielen der Sequenz n bewegt er sich wieder in Stellung H.



Legende: |||| = NF-Marke mit Nummer der folgenden Sequenz

↓ = Tonkopfstellung

Nach Vorwärtssuchen und Abspielen : Tonkopf rechts von der
entsprechenden Marke

Nach Rückwärtssuchen : Tonkopf links von der
entsprechenden Marke

Jeder Seitenwechsel im Suchlauf ergibt einen Impuls zu viel!

Fig. 3.17 : Die verschiedenen möglichen Tonkopfpositionen

Beim Uebergang von Vorwärtssuchen resp. Abspielen zu Rückwärtssuchen und umgekehrt muss der Tonkopf nun einen "Seitenwechsel" vornehmen. Erfolgt dieser Seitenwechsel im Suchlauf, so wird beim Ueberfahren der Marke n natürlich bereits ein Zählimpuls erzeugt. Dieser Seitenwechsel-Zählimpuls muss unterdrückt werden, denn man befindet sich ja immer noch bei Sequenz n. Erst die Marken n+1 resp. n-1 dürfen wieder gezählt werden.

Wenn im Suchlauf ein Seitenwechsel auftritt, muss man also in beiden Fällen den ersten Zählimpuls unterdrücken :

Fall 1 : Nach Vorwärtssuchen oder Abspielen folgt Rückwärtssuchen

Fall 2 : Nach Rückwärtssuchen folgt Vorwärtssuchen

Diese beiden Fälle werden durch die entsprechenden Korrekturstufen in Baugruppe KO erkannt und korrigiert (siehe Kap. 3.11.3).

3.11.3 Blockschema und Funktionsbeschreibung der Tonband-Steuerlektronik

Fig. 3.18 zeigt den prinzipiellen Aufbau der Tonband-Steuerlektronik.

Ein ankommender Tonbandsteuerbefehl wird mit Hilfe von CP5 in den Speicher 5 auf der Baugruppe TP eingelesen, wenn das Tonband frei ist ($RT = 1$), d.h. wenn der vorherige Suchprozess beendet ist. Da die digitalen Teile der Baugruppen ZV, KO, MD, und EV zur Erhöhung der Störsicherheit aus CMOS-Logik mit 12V Speisespannung aufgebaut sind, müssen die TTL-Signale anschliessend von einem Pegelwandler auf CMOS-Niveau gebracht werden. Bit 7, das Steuerbit Play (SP), wird direkt auf die Play-Kontrollogik in KO gegeben. Die Bits 6 - 0, die die Nummer der gewünschten Tonsequenz (Sollwert) darstellen, werden in ZV mit dem Inhalt des Positionszählers (Istwert) verglichen, welcher der aktuellen Stellung des Bandes entspricht.

Wenn der Sollwert grösser (kleiner) ist als der Istwert, so gibt der Komparator in ZV einen VOR (RUECK) - Befehl an das angeschlossene Tonbandgerät ab. Während des Suchlaufs registriert der Positionszähler laufend die Zählimpulse ZI, die aus den niederfrequenten Sequenzmarkierungen auf der Steuerspur gebildet werden. Auch die Zählrichtung wird vom Komparator gesteuert. Auf diese Weise entspricht der Zählerinhalt immer genau der momentanen Tonbandposition.

Die Rückmeldesignale des angeschlossenen Tonbandgerätes werden mit Tiefpässen und Schmitt-Triggern entstört. Wenn ein VOR-R oder RUECK-R Rückmeldesignal vorliegt, wird das Signal SU (Suchen) = 1. SU hat viele

wichtige Aufgaben. Es blockiert beispielsweise das Einlesen weiterer Tonbandsteuerbefehle während des Suchprozesses, indem es $RT = 0$ setzt.

Wenn beim Suchen Sollwert und Istwert gleich geworden sind, wird $GL = 1$ und somit auch $SS = 1$, wodurch ein STOP-Signal an das Tonbandgerät ausgegeben wird. Falls das Steuerbit Play $SP = 1$ ist, wird zudem $PA = 1$ und löst den PLAY-Impuls von 6s Dauer aus. Sobald das Band stillsteht, wird so die Wiedergabe der gewünschten Sequenz auf der Tonspur ausgelöst. Damit auch nach einem Rückwärtssuchen durch vorzeitiges Drücken der WEITER-Taste keine Fehler entstehen können, bleibt RT auch während der Dauer des PLAY-Impulses noch auf 0. Erst wenn die Sequenzmarkierungsimpulse sicher vorbei sind, geht RT wieder auf 1 und erlaubt so das Einlesen neuer Tonbandsteuerbefehle.

Nach dem Abspielen der Tonsequenz wird durch den hochfrequenten Impuls am Anfang der nächsten Sequenzmarkierung ein weiterer STOP-Impuls (HS) ausgelöst. Auf diese Weise hält das Band in genügendem Abstand vor der nächsten niederfrequenten Sequenzmarkierung. Bei einem eventuellen anschliessenden Vorwärtssuchen ist so eine ausreichende Beschleunigungsstrecke vorhanden, damit die Bandgeschwindigkeit für die sichere Detektion der nächsten NF-Marke ausreicht.

Nach dem Abspielen einer Tonsequenz ist $PA = 1$ und $STOP-R = 1$. Durch Drücken der Taste TONREPETITION kann die Sequenz beliebig oft wiederholt werden. Zu diesem Zweck wird ein Impuls ausgelöst, der den Positionszähler um 1 höher stellt. Die Steuerelektronik wird so auf einfache Weise veranlasst, das Tonband nochmals auf den Anfang der eben gehörten Sequenz zurückzuspulen und diese nochmals abzuspielen.

Ein Rücklaufbefehl für das Tonbandgerät kann nicht nur vom Komparator, sondern auch durch ein Reset-Signal ausgelöst werden. Das Reset-Flipflop in ZV wird unmittelbar nach dem Einschalten auf 1 gesetzt, ferner durch Drücken der Taste TONSTÖRUNG sowie wenn die Sollposition 0 verlangt wird und deshalb die Auswurferlaubnis $AE = 1$ ist. Reset setzt auch den Positionszähler auf 0 und hat noch verschiedene kleine Nebenaufgaben, die aus dem Blockschema nicht ersichtlich sind. Durch die abfallende Flanke eines SU -Signals wird das Reset-Flipflop wieder gelöscht.

Der Markendetektor MD hat die wichtige Aufgabe, aus den tonfrequenten Impulsen auf der Steuerspur unabhängig von äusseren Störeinflüssen immer genau einen CMOS-Impuls zu erzeugen.

Blockschema der Tonbandsteuerung

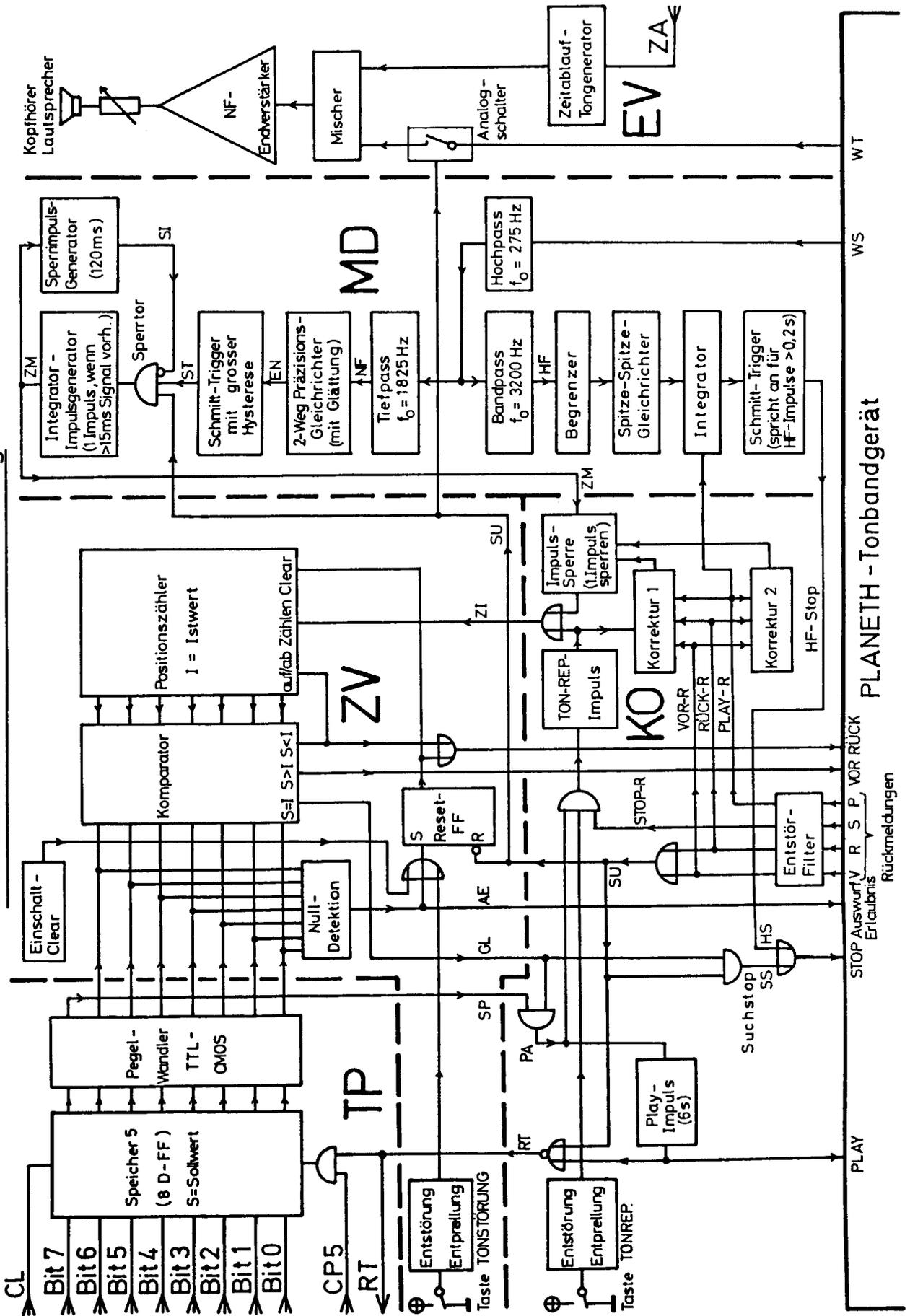


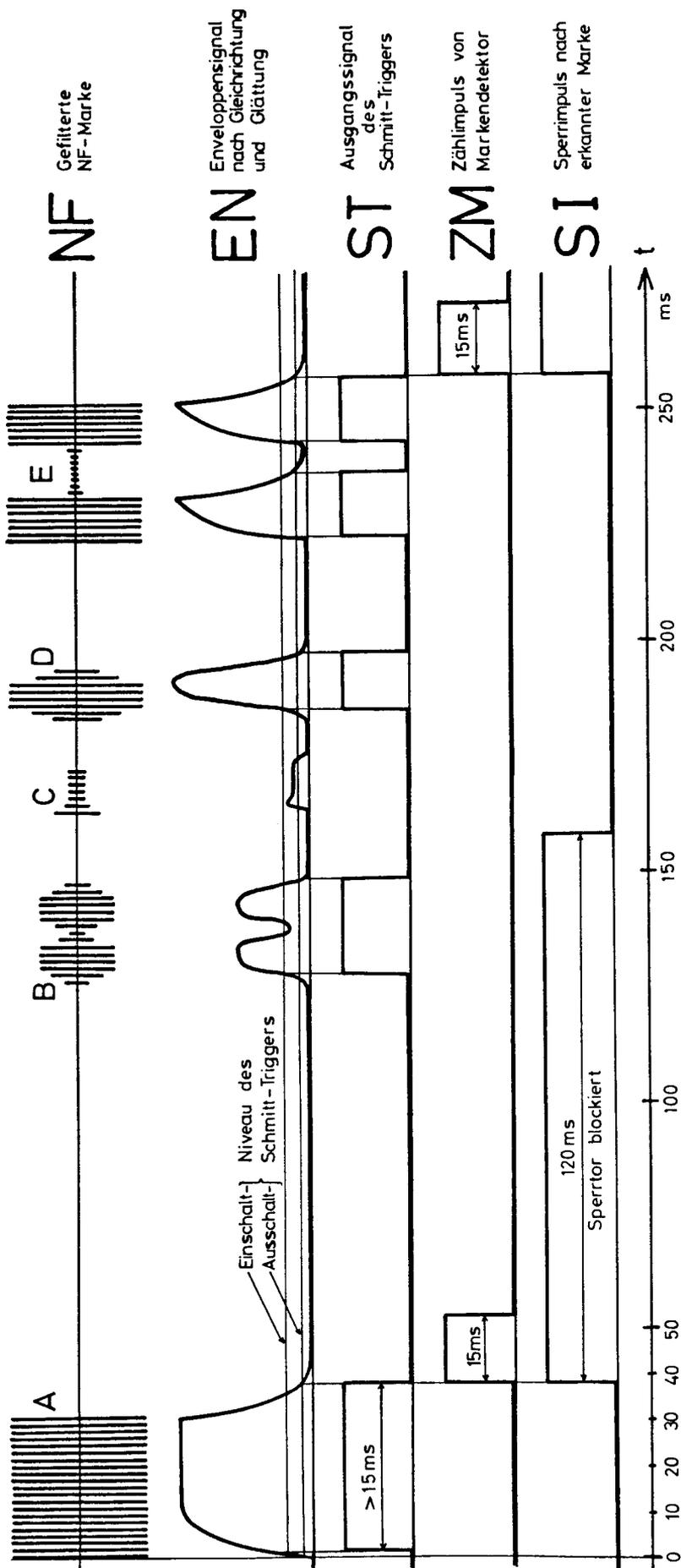
Fig. 3.18 : Blockschema der Tonbandsteuerung

Bei der Detektion der hochfrequenten Stopmarke ist dies nicht allzu schwierig. Da das Band mit der konstanten Wiedergabegeschwindigkeit läuft, ist die HF-Markenfrequenz (3,2kHz) konstant. Sie wird mit einem aktiven Bandpass ausgefiltert, begrenzt und gleichgerichtet. Das Signal wird darauf einem Integrator zugeführt, der jedoch nur arbeitet, wenn die PLAY-Rückmeldung bereits einige Zeit vorhanden ist. Wenn das HF-Signal mindestens 200ms lang gedauert hat, spricht ein Schmitt-Trigger an und gibt einen HF-Stopimpuls (HS) an die Baugruppe K0 ab.

Die Detektion der niederfrequenten 25Hz-Marken, die im Suchlauf gezählt werden müssen, wirft wesentlich mehr Probleme auf. Ihre Frequenz ist gemäss Formel (3.6) viel höher als die Aufnahme Frequenz. Ihre Dauer dagegen ist nach (3.7) viel kleiner als die Aufnahmedauer. Die normierte Suchgeschwindigkeit V_s ist nicht konstant, sondern kann von etwa 10 (unmittelbar nach Anlauf) bis 100 (bei schlecht eingestelltem Geschwindigkeitsregler im Tonbandgerät) variieren. Normalerweise ist V_s zwar auf etwa 45 (+/- 10%) geregelt, der Markendetektor muss jedoch auf die Extremwerte ausgerichtet sein. Infolge Bandflatterns und mechanischer Toleranzen der Kassetten können überdies Schwankungen des Kopf-Band-Abstandes und damit nach Formel (3.5) sehr starke Amplitudenschwankungen auftreten.

Fig. 3.19 zeigt die im NF-Markendetektor vorhandenen Signale. Das Signal der Steuerspur wird zur Unterdrückung des 1/f- und des Flickerrauschens zunächst über ein aktives Hochpassfilter 2. Ordnung ($f_0 = 275\text{Hz}$) geführt. Die niederfrequenten Markensignale gelangen anschliessend über einen aktiven Butterworth-Tiefpass 2. Ordnung [4] ($f_0 = 1825\text{Hz}$) auf einen 2-Weg Präzisionsgleichrichter mit Glättung. Ein Schmitt-Trigger mit besonders grosser Hysterese sorgt dafür, dass aus der so gewonnenen Enveloppe EN des Markierimpulses auch bei starken Amplitudenschwankungen möglichst nur ein Impuls ST entsteht. ST gelangt dann über ein Sperrtor, das nur im Suchlauf passiert werden kann, auf den Integrator-Impuls generator. Dieser gibt nach dem Verschwinden von ST einen Zählimpuls ZM ab, wenn ST ununterbrochen während mindestens 15ms vorhanden war. Fällt ST infolge von Amplitudenschwankungen kurzzeitig aus (Fall E in Fig. 3.19), so wird in den Impulspausen abintegriert, was zwar die notwendige Gesamtdauer für eine Anerkennung als zählender Impuls verlängert, diese jedoch noch nicht prinzipiell verunmöglicht. Jeder Zählimpuls triggert sofort den Sperrimpuls generator, der das Sperrtor mit dem Impuls SI

Die Signale des NF-Markendetektors in Funktion der Zeit



Legende: A: Normale NF-Marke, löst einen Zählimpuls ZM aus.

B: Störimpuls genügender Amplitude und Dauer, wegen noch vorhandenem Sperrimpuls SI aber kein ZM-Impuls.

C: Störimpuls zu geringer Amplitude, kein ST-Impuls und damit auch kein ZM-Impuls.

D: Störimpuls genügender Amplitude → ST-Impuls, jedoch zu kurz. Deshalb kein ZM-Impuls.

E: NF-Marke mit Amplitudeneinbruch → 2 aufeinanderfolgende ST-Impulse < 15ms. Markenerkennung dank der im Integrator gespeicherten Restladung vom 1. ST-Impuls. → 1 ZM-Impuls.

Fig. 3.19: Die Signale des NF-Markendetektors in Funktion der Zeit

während 120ms sperrt und so verhindert, dass aus einer NF-Marke infolge Amplitudenschwankungen und Drop-Outs zwei oder sogar mehr Zählimpulse erzeugt werden können. Dank all diesen Massnahmen funktioniert die Erkennung der NF-Marken sehr zuverlässig. Bis Ende März 1977 wurden in den Tonbandgeräten der PLANETH-Lehrkonsolen über 12000 Marken ausgezählt, wobei ein einziger Zählfehler auftrat.

Die so generierten Zählimpulse ZM dürfen aber noch nicht direkt dem Positionszähler zugeführt werden, sondern gelangen zuerst zur Impulssperre in der Baugruppe KO. Wie in Kap. 3.11.2 dargelegt, müssen bei bestimmten Folgen von Befehlen Korrekturen vorgenommen werden. Die beiden Korrekturstufen 1 und 2 registrieren die Vorgeschichte der einzelnen Befehle. Wenn sie eine kritische Befehlsfolge entdeckt haben, aktivieren sie die Impulssperre. Diese unterdrückt den ersten ankommenden Zählimpuls ZM, lässt die übrigen jedoch ungehindert durch.

In der Baugruppe EV wird das Signal der Tonspur aufbereitet. Den Analogschalter, der von SU gesteuert wird, kann es nur passieren, wenn kein Suchprozess im Gang ist. Dadurch werden unerwünschte Quietschgeräusche beim Suchen unterdrückt. Wenn die Zeit beinahe abgelaufen ist, wird $ZA = 1$, was den Zeitablauf-Tongenerator veranlasst, ein akustisches Warnsignal in den Tonkanal einzublenden. Das Tonsignal wird im NF-Endverstärker nochmals verstärkt und gelangt dann über einen Lautstärkereglern auf den angeschlossenen Kopfhörer.

3.12 Tastenspeicher (TA)

Für die Uebertragung der Information von den 16 Antworttasten zum Computer werden nur 4 der 8 Datenbits eines Tastenabfragebefehls benötigt, nämlich die Bits 3 - 0. Die übrigen 4 Bits (Bits 7 - 4) werden für die Sicherung der Datenübertragung eingesetzt, indem sie die gleichen Daten invertiert übertragen. Dadurch entsteht ein einfacher gleichgewichtiger Code, mit dem 1-Bit-Fehler sicher und Fehler bei mehreren Bits mit grosser Wahrscheinlichkeit erkannt werden können.

Taste	Bit	7	6	5	4	3	2	1	0
keine		0	0	0	0	0	0	0	0
0		1	1	1	1	0	0	0	0
1		1	1	1	0	0	0	0	1
2		1	1	0	1	0	0	1	0
3		1	1	0	0	0	0	1	1
4		1	0	1	1	0	1	0	0
5		1	0	1	0	0	1	0	1
6		1	0	0	1	0	1	1	0
7		1	0	0	0	0	1	1	1
8		0	1	1	1	1	0	0	0
9		0	1	1	0	1	0	0	1
-		0	1	0	1	1	0	1	0
,		0	1	0	0	1	0	1	1
START		0	0	1	1	1	1	0	0
STOP		0	0	1	0	1	1	0	1
WEITER		0	0	0	1	1	1	1	0
ZUSATZZEIT		0	0	0	0	1	1	1	1

Fig. 3.20 : Die Codierung der Antworttasten für die Datenübertragung

Dieses Verfahren ist natürlich von der Informationstheorie her gesehen nicht optimal. Mit einem einzigen Parity-Bit könnte man 1-Bit-Fehler auch erkennen und mit den verfügbaren 8 Bits könnte man sogar einen 1-Fehler-korrigierenden Code verwenden [1]. Die gewählte Methode benötigt jedoch sowohl hardwaremässig wie softwaremässig nur einen sehr geringen Aufwand und ist für die bereits sehr geringe Fehlerwahrscheinlichkeit bei der PLANETH-Datenübertragung völlig ausreichend. Fig. 3.20 zeigt die Codierung der Antworttasten für die Datenübertragung.

Fig. 3.21 zeigt den prinzipiellen Aufbau der Baugruppe Tastenspeicher. Ein Signal von einer Antworttaste wird zunächst mit einer speziellen Schaltung entprellt. Dank dieser Schaltung wird nur eine Verbindungsleitung (statt 2) pro Taste benötigt. Eventuelle Störimpulse werden mit einem Tiefpass ausgefiltert. Anschliessend gelangt das Signal zur Codierstufe. Wenn die Tasten nicht blockiert sind, d.h. wenn $WA = 0$ ist, erzeugt diese daraus den Code für die Datenübertragung, der in Speicher 7 zwischengespeichert wird. Gleichzeitig wird das INFO-Flipflop auf 1 gesetzt, was bedeutet, dass im Speicher 7 Tasteninformation gespeichert ist.

Das INFO-Signal bewirkt, dass die Tastensperrung aktiviert wird. WA geht auf 1 und weitere Tastensignale werden dadurch blockiert. Ausser durch INFO kann die Tastensperrung auch durch das Fehlen eines Rückmeldesignals RP, RL oder RT ausgelöst werden (von den Baugruppen DS, LS resp. TP, wenn Projektoren im Suchlauf, WARTEN-Lampe gewünscht oder Tonband im Suchlauf).

Damit bei der Uebertragung auf der Party-Line keine Fehler auftreten, müssen die Konsolen-Ausgangssignale unabhängig vom Zeitpunkt des Tastendrucks immer die in Kap. 3.4.3 angegebene Form haben. Deshalb sorgt die Synchronisierstufe dafür, dass bei jedem Tastenabfragebefehl nur die Information ausgegeben werden kann, die beim Beginn des Abfragebefehls bereits im Speicher 7 vorhanden war. Wird eine Taste während eines Abfragebefehls gedrückt, so kann die entsprechende Information erst beim nächsten Abfragebefehl zum Computer gelangen.

Sobald die Konsolensteuerung KS einen Konsolenabfragebefehl feststellt, geht das Steuersignal S7 auf 1. Wenn INFO bei diesem Anstieg von S7 bereits 1 ist, veranlasst die Synchronisierstufe das Oeffnen der Ausgangstore. Die Daten in Speicher 7 gelangen so auf die Ausgangsdatenleitungen Bit 3 - 0. Auf Bit 7 - 4 sind die gleichen Daten invertiert vorhanden. So gelangt die Tasteninformation über die Driverschaltungen in PD auf die Party-Line-Datenbitleitungen und von dort zum Computer.

Wenn der Takt CP7 wieder auf 0 geht, wird der Speicher 7 und das INFO-Flipflop gelöscht. Sofern keine äusseren Blockiersignale vorliegen (RP, RL oder RT auf 0), werden darauf die Tasten wieder für weitere Eingaben freigegeben.

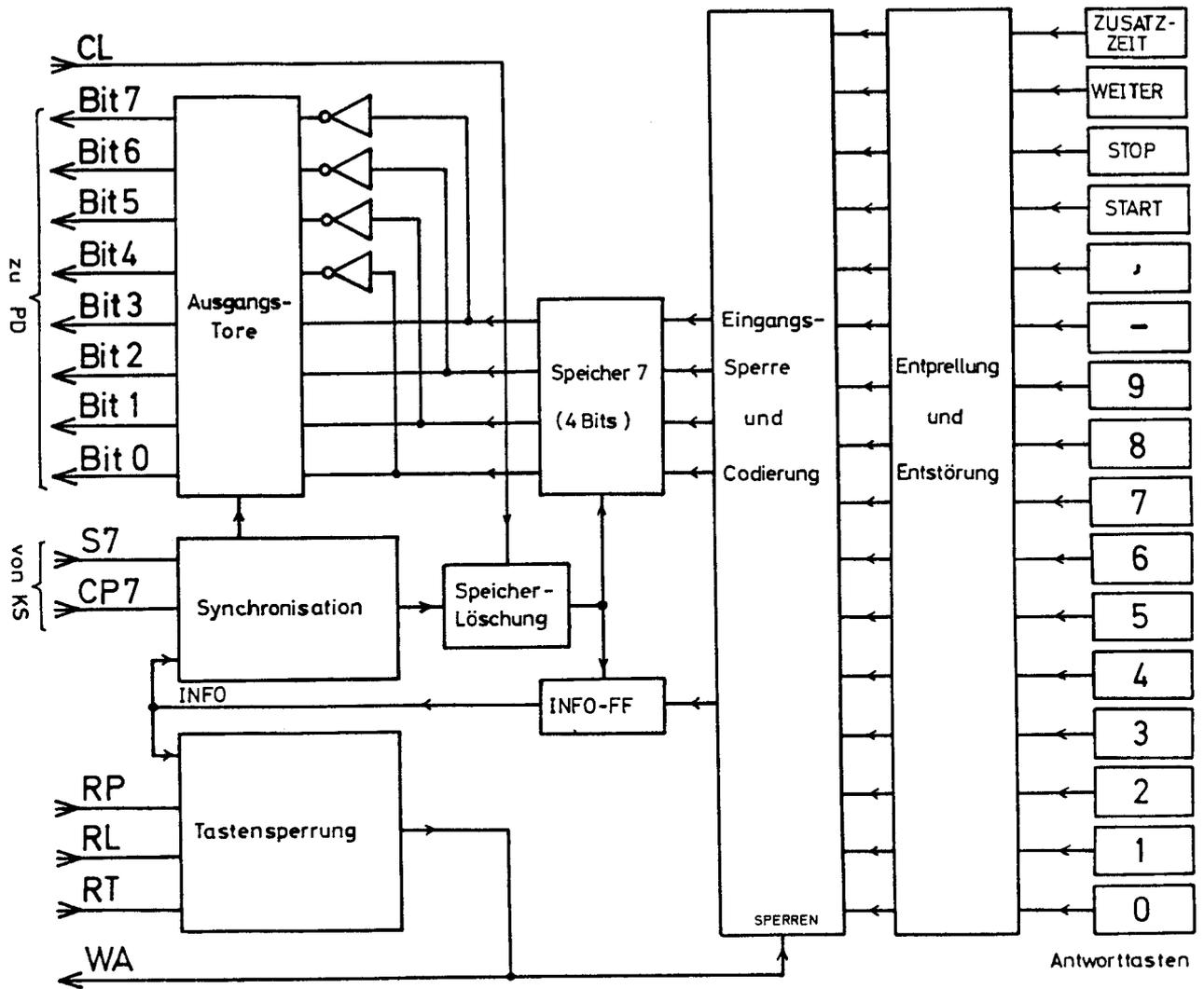


Fig. 3.21 : Blockscheema der Baugruppe Tastenspeicher (TA)

4. DIE GERAETE DER ZENTRALEN STEUERUNG

Für das Verständnis des Lehrsystems allgemein und der Computerprogramme im besonderen ist auch ein Ueberblick über die wichtigsten technischen Daten der Funktionseinheiten erforderlich, die in der zentralen Steuerung verwendet werden. Dabei sollen der Computer und die serienmässigen Peripherieeinheiten nur kurz besprochen werden. Das Vorschaltgerät, das die Pegelwandlung zwischen Computer und Party-Line besorgt, wird danach etwas ausführlicher behandelt.

4.1 Der Computer 2116B von Hewlett-Packard

Der Computer ist aus integrierten Schaltungen aufgebaut (CTL*, heute weniger gebräuchlich, und teilweise TTL). Er besitzt 7 Arbeitsregister, davon 2 eigentliche Rechenregister zu je 16 Bits. Der Kernspeicher umfasst 8K (8192) Worte zu 16 Bits. Er ist in sogenannte "Seiten" von je 1K (1024) Worten aufgeteilt. Ausser auf der untersten Seite (Base Page), die von überall her direkt adressierbar ist, können auf jeder Seite nur die Speicherzellen auf der gleichen Seite (Current Page) direkt adressiert werden. Für Verknüpfungen zwischen den Seiten muss die indirekte Adressierung verwendet werden.

Der Computer ist mit einer Interruptlogik ausgerüstet, die einen Interruptbetrieb mit maximal 48 peripheren Geräten zulässt. Die Priorität der Interrupt-Verarbeitung ist dabei rein hardwaremässig durch die Schlitznummer der Interface-Karte festgelegt, an die das entsprechende Gerät angeschlossen ist. Je niedriger diese Nummer ist, desto höher ist die Priorität.

Der Computer ist mit einem zusätzlichen Rechenwerk ausgerüstet, das die direkte hardwaremässige Durchführung von Multiplikationen und Divisionen mit ganzen Zahlen gestattet. Dadurch wird die sonst für Multiplikationen und Divisionen mit Unterprogrammen benötigte Zeit ganz wesentlich verkürzt.

*) CTL = Komplementär-Transistor-Logik [4]

Technische Daten des hp-2116B-Computers :

Speicherkapazität : 8K 16-Bit-Worte (ausbaubar im Originalgehäuse auf 16K,
maximal bis auf 32K)

Register : 7 , davon 2 eigentliche Rechenregister

Wortlänge : 16 Bits (Befehle für Zusatzrechenwerk: 2·16 Bits)

Befehlsvorrat : 70 Grundinstruktionen
11 Befehle für Zusatzrechenwerk (EAU-Option)

Adressierbarkeit : Direkt auf unterster und laufender Seite (1 Seite = 1K),
indirekt (auch mehrfach) im ganzen Speicher

Taktfrequenz : 10 MHz

Zykluszeit : 1,6µs

Operationszeiten : Ganze Zahlen (-32768 bis +32767):
Addition : 3,2µs
Multiplikation : 19,2µs
Division : 20,8µs
Reelle Zahlen (Betrag 10^{-38} bis 10^{+38} und 0):
Addition : 900µs }
Multiplikation : 750µs } mit Unterprogrammen
Division : 1500µs }

Anschliessbare
Peripheriegeräte : Direkt 16 Stück (maximal auf 48 erweiterbar)

4.2 Die am Computer angeschlossenen serienmässigen Ein- und Ausgabegeräte

4.2.1 Fernschreibmaschine

Sende- und Empfangs-Terminal mit alphanumerischer Tastatur und Druckwerk
Typ : Teletype 33 ASR mit automatischem Motorstart
Schreibgeschwindigkeit : 10 Zeichen pro Sekunde
Schreibbreite : 72 Zeichen pro Linie
Zeichensatz : ASC II

Das Gerät kann auch zum langsamen Lesen und Stanzen von Lochstreifen eingesetzt werden. Das PLANETH-System nützt jedoch diese Möglichkeiten wegen der viel zu geringen Verarbeitungsgeschwindigkeit nicht aus.

4.2.2 Lochstreifenleser

Typ : hp 2737A (Mechanik von REMEX)
Lesegeschwindigkeit : 250 Bytes* pro Sekunde
Lochstreifen : 1 Zoll (25,4mm) breit, 8 Spuren
Startzeit : 6ms
Stopzeit : 1ms

4.2.3 Lochstreifenstanzer

Typ : Facit 4070
Stanzgeschwindigkeit: 75 Bytes* pro Sekunde
Lochstreifen : 1 Zoll (25,4mm) breit, 8 Spuren

4.3 Das Party-Line-Vorschaltgerät

Das Party-Line-Vorschaltgerät führt die Pegelwandlung zwischen den Signalniveaus des Computers (TTL, 0V/3V) und der Party-Line (0V/24V) durch. Neben der Pegelwandlung zeigt es zur Ueberwachung des Datenverkehrs alle Signale auf der Party-Line laufend mit Leuchtdioden an. Dadurch wird auch das Austesten der ganzen Anlage wesentlich erleichtert. Das Vorschaltgerät bildet aus der Netzfrequenz zudem einen Zeittakt für die innere Uhr des PLANETH-Systemprogramms. Ferner erzeugt es die für den Betrieb der Party-Line benötigten Spannungen von +/-24V (siehe Kap. 3.4.2), überwacht gewisse elementare Funktionen der ganzen Anlage und gibt Signale zur Fernanzeige des Systemstatus aus.

Fig. 4.1 zeigt das Blockschema des Vorschaltgerätes. Auf der Computerseite ist es an eine serienmässige Interfacekarte hp 12566A angeschlossen, welche die parallele Ein- und Ausgabe von je 16 Bits nebst Ausgabetak (Control) und Eingabetakt (Flag) erlaubt. Alle Signale sind TTL-kompatibel (logisch 1 \approx 0V, logisch 0 \approx 3V).

Bei jedem Bit wird das vom Computer kommende Signal mit einer nicht-invertierenden Sendeschaltung S (siehe Fig. 3.7) auf das Spannungsniveau der Party-Line (0V/24V) gebracht. Der Widerstand $R_p = 1,5k\Omega$ zieht die Signalader bei abgeschalteter Sendeschaltung (logisch 0) auf +24V. Eine invertierende Empfangsschaltung E (siehe Fig. 3.7)

*) 1 Byte = 8 Bits

Blockschema des Party-Line-Vorschaltgerätes

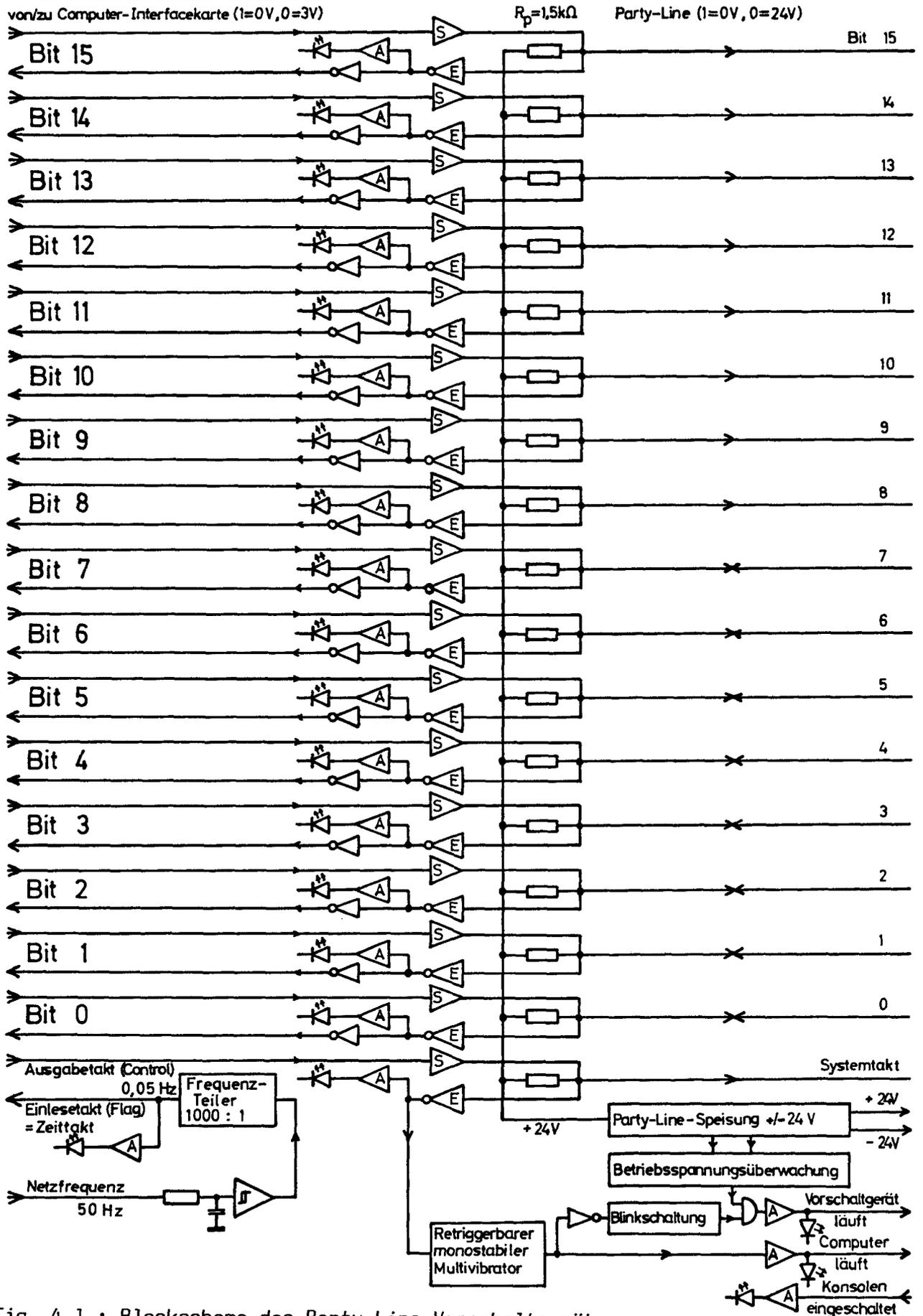


Fig. 4.1 : Blockschema des Party-Line-Vorschaltgerätes

wandelt das Signal auf der Party-Line wieder in ein TTL-Signal um. Ueber einen Inverter gelangt es zum Computer. Mit einer Treiberschaltung A wird eine Leuchtdiode angesteuert, die das Party-Line-Signal laufend anzeigt. Eine ausführlichere Beschreibung der Sende-/Empfangsschaltungen und der Signalverläufe auf der Party-Line befindet sich in Kap. 3.4 .

Eigentlich werden im Vorschaltgerät die Empfangsschaltungen E nur für die 8 Datenbits (Bits 7 - 0) benötigt, da nur auf diesen Adern auch Signale von den Konsolen zum Computer fliessen. Die Verwendung von Empfangsschaltungen bei allen Bits erlaubt aber eine ständige Ueberwachung der Party-Line durch den Computer, was die Zuverlässigkeit der Datenübertragung wesentlich erhöht.

Als Systemtakt, der den Zeitpunkt der Datenübernahme steuert, wird der Ausgabetakt (Control) verwendet. Das Signal von der Systemtakt-Empfangsschaltung triggert laufend einen retriggerbaren monostabilen Multivibrator. Solange im Computer das richtige Programm läuft und das Vorschaltgerät richtig funktioniert, werden ständig Systemtaktimpulse auf die Party-Line ausgegeben. Wenn auf der Systemtaktader keine Störung auftritt, bleibt das Ausgangssignal dieses Multivibrators somit dauernd auf 1 und zeigt dadurch an, dass der Computer ordnungsgemäss funktioniert. Ueber eine Treiberschaltung A wird damit eine grüne Leuchtdiode "Computer läuft" angesteuert. Das Ausgangssignal dieses Treibers wird zudem noch auf eine Statussignalader der Party-Line gegeben, damit der Computerbetriebszustand auch bei den Lehrkonsolen und im Raum des verantwortlichen Assistenten angezeigt werden kann.

Für das Einlesen der von der Party-Line kommenden Daten wird kein Einlesetaktimpuls benötigt, denn der Datenfluss auf der Party-Line wird nur durch das Computerprogramm und die von diesem erzeugten Systemtaktimpulse gesteuert (siehe Fig. 3.8). Die Einlesetaktimpuls-Leitung (Flag) kann deshalb für die Eingabe zusätzlicher Information ausgenutzt werden. Für die innere Uhr des PLANETH-Systemprogramms wird ein Zeittakt-Signal benötigt. Zu diesem Zweck wird die 50Hz-Netzfrequenz zunächst gefiltert und dann auf einen Schmitt-Trigger geleitet. Das entstehende 50Hz-Rechtecksignal wird einem 1000:1 Frequenzteiler zugeführt. Sein Ausgangssignal hat noch eine Frequenz von 0,05Hz und wechselt seinen Zustand alle 10 Sekunden. Dieses Signal wird als Zeittakt über die Einlesetakt-(Flag-)Leitung in den Computer eingespeist.

Das Vorschaltgerät erzeugt auch die für den Betrieb der Party-Line erforderlichen Spannungen von +24V und -24V (siehe Kap. 3.4.2). Die Spannung von -24V wird von den Empfangsschaltungen zur Erzielung des gewünschten Umschaltniveaus von 9,8V zwischen den Zuständen 0 und 1 benötigt. Zudem dient sie zur Verbesserung der Sperreigenschaften von abgeschalteten Sende- und Empfangstransistoren. Die Spannung von +24V speist die Hilfschaltungen in den Konsolen, die beim Ein- und Ausschalten die Ausgabe unkontrollierter Daten auf die Party-Line verhindern. Da die benötigten Stromstärken in all diesen Fällen sehr klein sind, ist eine zentrale Versorgung der Konsolen mit diesen Spannungen zweckmässig.

Die Betriebsspannungen von Party-Line und Vorschaltgerät werden von einer weiteren Baugruppe laufend überwacht. Solange all diese Spannungen auf ihrem Sollwert sind, wird über eine Treiberschaltung A eine orangefarbige Leuchtdiode "Vorschaltgerät läuft" angezündet. Auch dieses Signal wird zwecks Fernanzeige noch auf eine Statussignalader der Party-Line geleitet.

Wenn das Vorschaltgerät richtig funktioniert, der Computer jedoch nicht, so wird eine Blinkschaltung aktiviert, welche ein rythmisches Blinken der Leuchtdiode "Vorschaltgerät läuft" auslöst, damit man eher auf diesen Fehler aufmerksam wird.

Jede eingeschaltete Konsole gibt ein Signal von etwa 5V auf eine dritte Statussignalader. Wenn dieses Signal vorhanden ist, so zündet eine weitere Treiberschaltung im Vorschaltgerät eine rote Leuchtdiode "Konsolen eingeschaltet" an. Auch bei diesem Signal ist wie bei den andern Statussignalen eine Fernanzeige an verschiedenen Orten möglich.

Dank diesen drei fernangezeigten Statusinformationssignalen kann man überall längs der Party-Line sofort einen Ueberblick über den Betriebszustand des Lehrsystems PLANETH gewinnen. Dadurch wird der Betrieb und die Ueberwachung der Anlage sehr erleichtert.

5. DAS PLANETH - SYSTEMPROGRAMM

Die in Kap. 3 beschriebenen PLANETH-Lehrkonsolen können nur Befehle des zentralen Computers ausführen, also z.B. ein bestimmtes Dia zeigen, eine bestimmte Tonsequenz abspielen oder eine Information über die zuletzt gedrückte Antworttaste an den Computer abgeben.

Damit überhaupt ein Lehrbetrieb möglich ist, muss im steuernden Computer ein entsprechendes Programm laufen. Dieses Programm ist das PLANETH-Systemprogramm, das gut die Hälfte des 8K-Kernspeichers des Computers belegt (siehe Fig. 5.1). Es steuert den Datenaustausch zwischen Computer und Konsolen und sorgt dafür, dass auf jeder Konsole das gewünschte Lehrprogramm abläuft.

Das Systemprogramm benötigt dazu eine Anzahl Statusinformationen von jeder Konsole, in denen Angaben über das Verhalten des Adressaten und über den aktuellen Stand im Lehrprogramm vorhanden sind. Diese Statusinformationen sind in den sogenannten Konsolen-Status-Parametern gespeichert. Nähere Angaben in Kap. 5.3 .

Die Informationen über die Struktur und den Ablauf der einzelnen Lehrprogramme (= Lehrprogramme im engeren Sinn) sind im Rest des Speichers, dem Lehrprogramm Speicher, untergebracht. Der zugehörige Lehrstoff, der sehr viel mehr Computerspeicherplatz benötigen würde, ist dagegen auf den Dias und den Tonkassetten in den einzelnen Konsolen gespeichert. Der Lehrprogramm Speicher ist dabei optimal ausgenützt, d.h. jedes Lehrprogramm belegt nur den seiner wirklichen Grösse entsprechenden Platz. Je nach ihrer Länge können etwa 6 - 16 Lehrprogramme geladen werden.

Die Lehrprogramme können während des Betriebs nach Bedarf ganz oder teilweise gegen andere ausgetauscht werden, ohne dass irgendwelche Störungen oder Unterbrüche auftreten. Dadurch ist das System sehr flexibel und kann sich auch einer rasch ändernden Nachfrage nach Lehrprogrammen problemlos anpassen.

Bei der Entwicklung der PLANETH-Software wurde das Schwergewicht auf eine möglichst einfache, betriebssichere Programmierung ohne komplizierte, aber eigentlich unnötige Feinheiten gelegt. Eine Abschätzung der Systembelastung (siehe Kap. 5.2) zeigte, dass für einen Anschluss von maximal 32 Konsolen die Verwendung von Interrupt-Unterprogrammen

Die Belegung des 8K-Kernspeichers durch das PLANETH-Betriebssystem

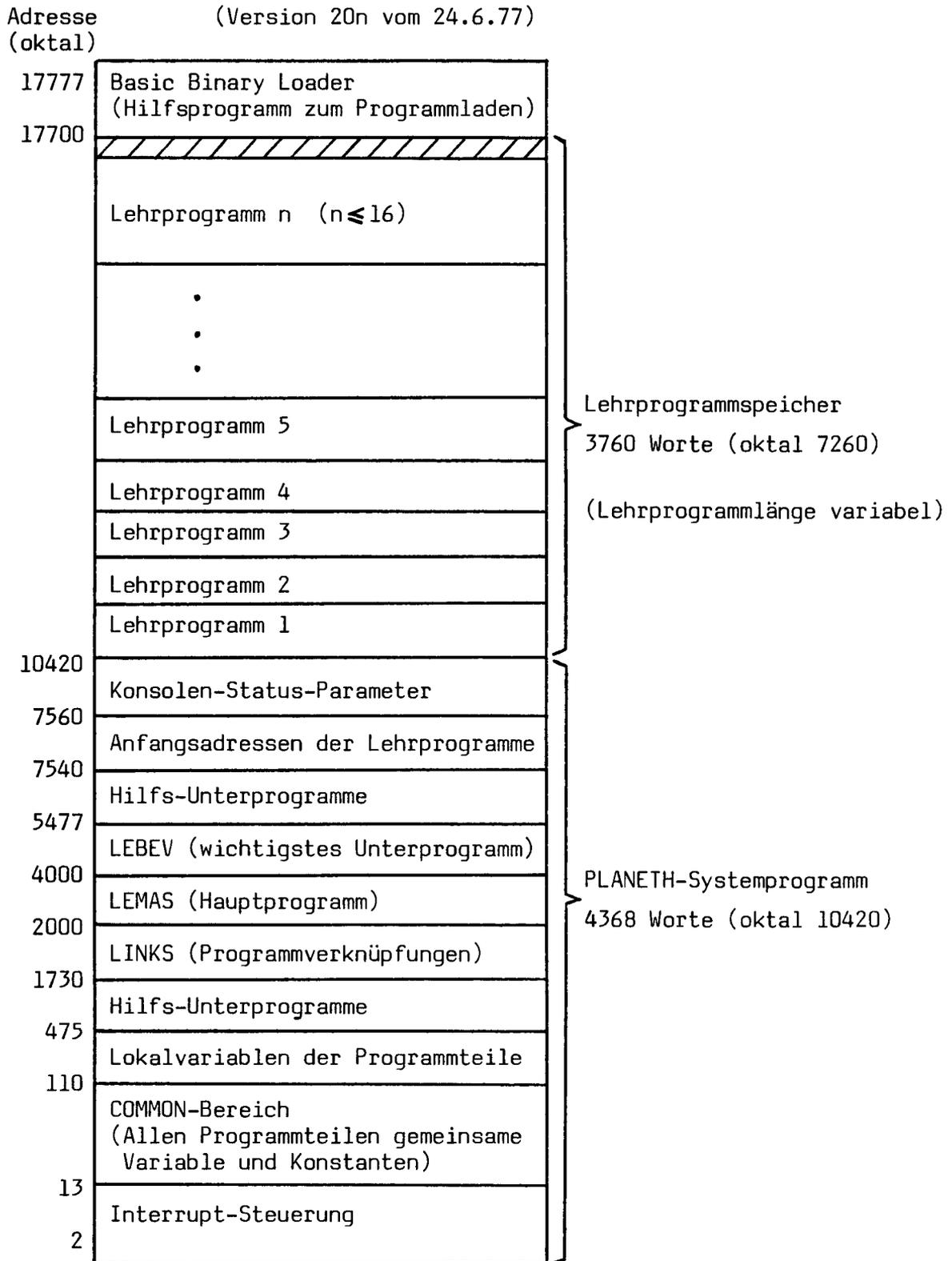


Fig. 5.1 : Die Belegung des Kernspeichers durch das PLANETH-Betriebssystem

für die Ein- und Ausgabe in den meisten Fällen unnötig war. Nur für das Ausdrucken der Kurzprotokolle auf der Fernschreibmaschine (Dauer ca. 21,5s) am Schluss jedes Lehrprogrammdurchlaufs wurde ein Warteschlangen-Ausgabe-Unterprogramm mit Interrupt entwickelt, bei dem die Konsolenbenützer auch beim Auftreten einer Warteschlange immer sofort bedient werden.

Das PLANETH-Systemprogramm wurde in ASSEMBLER geschrieben und für möglichst geringen Speicherplatzbedarf ausgelegt. Natürlich musste dafür etwas Rechengeschwindigkeit geopfert werden. Diese ist jedoch selbst bei Vollauslastung mit 32 Konsolen noch so gross, dass nur unmerkliche Systemreaktionszeiten auftreten (siehe Kap. 5.2.2). Auch die Länge der einzelnen Programmteile wurde so optimiert, dass für das Zusammenfügen der Teile zum ganzen Programm möglichst wenig Indirekt-Sprünge (LINKS) benötigt werden.

Das PLANETH-Systemprogramm wurde soweit als möglich modular aufgebaut, d.h. in verschiedene Programmteile und Unterprogramme aufgeteilt, die jeweils für bestimmte spezifische Aufgaben zuständig sind. Dies erleichtert die Uebersicht über den Aufbau und die Funktion des Programms sehr. Auch nachträgliche Aenderungen lassen sich dank dieser Konzeption leicht durchführen.

5.1 Grobflussdiagramm und prinzipielle Funktion des PLANETH-Systemprogramms

Die prinzipielle Struktur des PLANETH-Systemprogramms ist relativ einfach. Nach dem Start des Programms werden zunächst die benötigten Lehrprogramme eingelesen. Dann gelangt das Programm in die sogenannte Hauptschleife. Darin werden alle Konsolen zyklisch auf Tasteninformation abgefragt. Wenn eine Taste gedrückt wurde, wird die Hauptschleife kurz verlassen, damit die nötigen Massnahmen getroffen werden können. Danach erfolgt wieder ein Rücksprung in die Hauptschleife.

Fig. 5.2 zeigt das Grobflussdiagramm des PLANETH-Systemprogramms. Nach dem Einlesen der gewünschten Lehrprogramme mit dem Lochstreifenleser wird die Nummer der betrachteten Konsole zunächst 0 gesetzt.

Großflussdiagramm des PLANETH-Systemprogramms

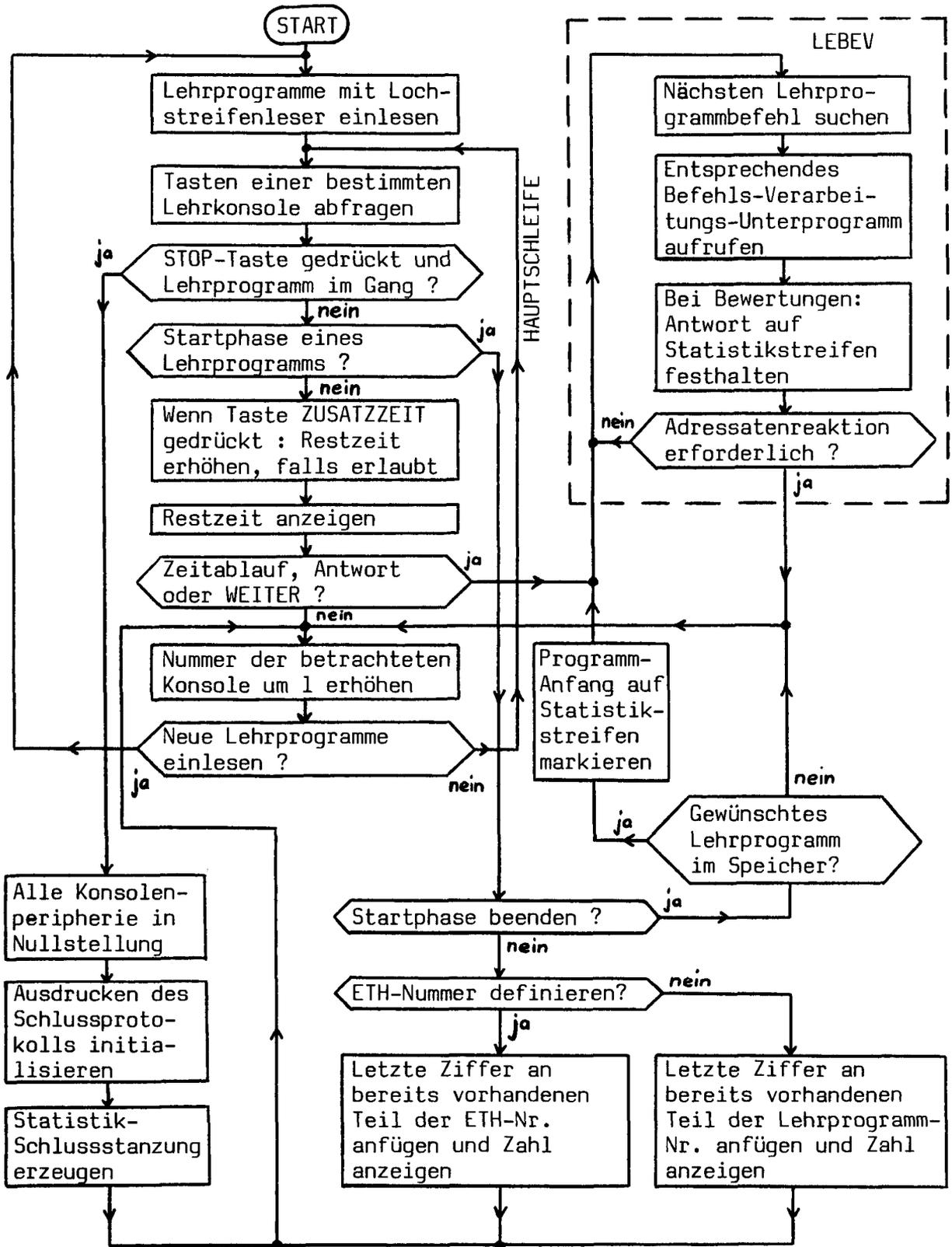


Fig. 5.2 : Großflussdiagramm des PLANETH-Systemprogramms

Am Anfang der Hauptschleife wird der Tastenspeicher der betrachteten Konsole abgefragt. Wenn ein Programm im Gang ist und die STOP-Taste gedrückt wurde, so werden zunächst alle Einheiten der Konsolenperipherie (Projektoren, Tonband usw.) auf 0 zurückgesetzt. Dann wird in der Ausgabewarteschlange der Fernschreibmaschine eingetragen, dass für die betreffende Konsole ein Schlussprotokoll ausgedruckt werden soll. Schliesslich wird auf dem Statistiklochstreifen, auf dem alle Antworten laufend ausgestanzt werden, der Schluss des Programmdurchlaufs markiert. Anschliessend erfolgt die Rückkehr in die Hauptschleife und die Vorbereitung der Abfrage der nächsten Konsole.

Wenn dagegen auf der betrachteten Konsole noch kein Lehrprogramm im Gang ist, befindet man sich in der sogenannten Startphase. In ihr muss zuerst die 8-stellige ETH-Nummer und darauf die 4-stellige Programmnummer eingegeben werden. Die zuletzt eingetippte Ziffer wird an den bereits eingegebenen Teil der entsprechenden Nummer angefügt und die neue Zahl auf die numerische Anzeige in der Konsole ausgegeben. Wenn die beiden Nummern noch nicht vollständig eingegeben sind, erfolgt wieder ein Rücksprung in die Hauptschleife und die Vorbereitung der Abfrage der nächsten Konsole. Falls jedoch beide Nummern vollständig eingegeben worden sind und das gewünschte Lehrprogramm im Speicher vorhanden ist, wird auf dem Statistikstreifen der Anfang eines neuen Lehrprogrammdurchlaufs markiert. Anschliessend erfolgt ein Sprung in das wichtige Unterprogramm LEBEV, das die ersten Befehle des gewünschten Programms ausführt.

Oft ist jedoch auf der betrachteten Konsole bereits ein Lehrprogramm im Gang und es wurde nicht die STOP-Taste gedrückt. Die Zeit, während der eine bestimmte Bildkombination gezeigt oder eine bestimmte Tonsequenz abgespielt wird, ist meist vom Lehrprogramm her beschränkt. Wenn die Taste ZUSATZZEIT gedrückt wird, wird diese Zeit erhöht, falls dies vom Autor des Lehrprogramms erlaubt wurde (Details siehe Kap. 6). Anschliessend wird die Restzeit, d.h. die Zeit, die noch bleibt, bis das Lehrprogramm automatisch weiterfährt, auf die Zeitanzeigeeinheit in der Konsole ausgegeben.

Wenn die Zeit abgelaufen ist, eine Antwort eingegeben worden ist oder die WEITER-Taste gedrückt worden ist, erfolgt ein Sprung ins Unterprogramm LEBEV, das die nächsten Lehrprogrammbefehle ausführt.

Wurde dagegen überhaupt keine Taste gedrückt, wird die Abfrage der nächsten Konsole vorbereitet, indem die Nummer der betrachteten Konsole um 1 erhöht wird, wobei die Nummer 32 wieder = 0 gesetzt wird. Darauf wird getestet, ob am Schalterregister des Computers das Einlesen oder der Austausch von Lehrprogrammen gewünscht wird. In diesem Fall erfolgt ein Sprung an den Programmanfang. Normalerweise muss jedoch keine Veränderung im Lehrprogramm-speicher vorgenommen werden, so dass wieder an den Anfang der Hauptschleife gesprungen werden kann.

Im wichtigen Unterprogramm LEBEV (Lehrprogramm-Befehls-Verarbeitung) muss zuerst der nächste Lehrprogrammbefehl im Lehrprogramm-speicher gesucht werden. Anschliessend wird das entsprechende interne Unterprogramm für die Verarbeitung dieses Befehls aufgerufen. Bei Bewertungsbefehlen wird die eingegebene Antwort für eine spätere statistische Analyse auf dem Statistiklochstreifen festgehalten. Wenn ein Befehl ausgeführt wurde, der eine Reaktion des Adressaten erfordert, erfolgt wieder ein Rücksprung in die Hauptschleife und die Vorbereitung der Abfrage der nächsten Konsole. Andernfalls wird gleich der nächste Lehrprogrammbefehl gesucht und ausgeführt.

Die vorstehende Beschreibung konnte natürlich nur einen ziemlich allgemeinen Ueberblick über die Funktion des PLANETH-Systemprogramms geben. Viele Einzelheiten und Spezialfälle konnten noch nicht behandelt werden. Genauere Angaben finden sich in den folgenden Kapiteln.

5.2 Abschätzung der Reaktionszeiten des PLANETH-Systemprogramms

Aus didaktischen Gründen ist nach einer Aktion des Adressaten an der Lehrkonsole eine möglichst rasche Reaktion des Lehrsystems erwünscht. Diese soll wenn möglich noch in der Gegenwartsdauer erfolgen, also innerhalb der Zeit, in welcher dem Adressaten seine letzte Handlung auch ohne spezielle Konzentration darauf noch bewusst bleibt. Wartezeiten bis etwa 6 Sekunden sind in der Regel pädagogisch noch tragbar [13].

Die Zugriffszeiten zu den Lehrstoffspeichern (Diaprojektoren, Tonbandgerät) betragen jeweils einige Sekunden (Mittelwert ca. 3,5s) und liegen somit in der oben erwähnten Grössenordnung. Wenn der Adressat jedoch nach jedem Tastendruck sofort (d.h. innerhalb von höchstens $\frac{1}{2}$ s) eine erste Rückmeldung vom Lehrsystem erhält (z.B. Anzünden der RICHTIG- oder der FALSCH-Lampe),

so ist der Einfluss dieser Wartezeiten sehr viel geringer. Das Lehrprogramm muss also in der Lage sein, jeden Tastendruck an irgend einer Konsole innerhalb von $\frac{1}{2}s$ zu verarbeiten.

Wenn mehrere Adressaten an verschiedenen Konsolen kurz hintereinander oder sogar gleichzeitig eine Taste drücken, können sogenannte Warteschlangen entstehen, weil das System nur eine Konsole auf einmal bedienen kann. Eine Untersuchung dieses Problems bei einem etwas einfacheren Lehrsystem wurde bereits in [5] durchgeführt.

5.2.1 Kurze Behandlung des Warteschlangenproblems mit einer Bedienstationsstation

Im einfachsten Fall - und dieser liegt hier vor - besteht das zu untersuchende Wartesystem aus einer Bedienstationsstation (dem Computer) und mehreren gleichartigen Einheiten (den Lehrkonsolen), die gelegentlich die Dienste der Bedienstationsstation in Anspruch nehmen. Sowohl die zeitliche Folge der Aufrufe der Bedienstationsstation als auch der Zeitbedarf für die Bedienung sind stochastisch, d.h. sie gehorchen einer Wahrscheinlichkeitsverteilung. Da die Bedienstationsstation immer nur einen Aufruf auf einmal bearbeiten kann, müssen eventuell weitere eintreffende Aufrufe bis zu ihrer Bearbeitung in eine Warteschlange eingereiht werden.

Man interessiert sich bei solchen Warteschlangen-Problemen für die Wahrscheinlichkeit, dass eine Einheit warten muss und für die durchschnittliche Länge der entstehenden Warteschlange. Aus diesen Angaben kann man Rückschlüsse ziehen auf die notwendige Leistungsfähigkeit der Bedienstationsstation und auf die Zahl der maximal anschliessbaren Einheiten.

Man bezeichnet als mittlere Ankunftsrate a die durchschnittliche Anzahl Aufrufe der Bedienstationsstation pro Zeiteinheit. Die mittlere Bedienungsrate b ist dagegen die durchschnittliche Anzahl Einheiten, die in einer Zeiteinheit bedient werden können. Die Grösse $\psi = a/b$ schliesslich wird als Verkehrsintensität bezeichnet.

Nach [12] gelten im Falle zufälligen Zugangs und Abgangs für die Wahrscheinlichkeit einer bestimmten Warteschlangenlänge und für die durchschnittliche Warteschlangenlänge folgende Beziehungen :

1. Wahrscheinlichkeit einer Warteschlange der Länge n :

$$P_n = \psi^n(1 - \psi) \quad , \text{ wenn } \psi < 1 \quad (5.1)$$

2. Durchschnittliche Länge einer Warteschlange :

$$\bar{n} = \frac{\psi}{1 - \psi} \quad , \text{ wenn } \psi < 1 \quad (5.2)$$

wobei $\psi = a/b = \text{Verkehrsintensität} = \frac{\text{Anzahl Aufrufe pro Zeiteinheit}}{\text{Anzahl bediente Einheiten pro Zeiteinheit}}$

$n = \text{Anzahl Einheiten in der Warteschlange (inkl. der gerade bedienten)}$

Die angegebenen Beziehungen gelten unter der Voraussetzung, dass die Werte für a und b und damit auch für ψ für den zu betrachtenden Zeitraum als konstant angesehen werden können. Es wird weiter angenommen, dass die Bedienungsrate b von der Anzahl der wartenden Einheiten unabhängig ist und dass die Einheiten, welche die Warteschlange bilden, in der Reihenfolge ihres Eintreffens abgefertigt werden.

Aus (5.1) kann man mit den bei geometrischen Reihen gültigen Beziehungen ableiten :

3. Wahrscheinlichkeit $P_{n > m}$, dass die Warteschlange eine Länge $> m$ aufweist :

$$P_{n > m} = \sum_{n=m+1}^{\infty} P_n = (1 - \psi)\psi^{m+1} \sum_{n=0}^{\infty} \psi^n = \psi^{m+1} \quad (5.3)$$

4. Wahrscheinlichkeit $P_{n \leq m}$, dass die Warteschlange eine Länge $\leq m$ aufweist :

$$P_{n \leq m} = 1 - P_{n > m} = 1 - \psi^{m+1} \quad (5.4)$$

Für $\psi > 1$ ist die Bedienstation überfordert, die Warteschlange wächst unbegrenzt an und die Beziehungen (5.1) bis (5.4) sind nicht mehr gültig.

5.2.2 Die Reaktionszeit bei der Verarbeitung von Tasteninformation

Für die Abfrage einer Konsole im Leerlauf, d.h. wenn keine Tasteninformation verarbeitet werden muss, sind etwa $400\mu\text{s}$ erforderlich. Die Differenz gegenüber der in Fig. 3.8 angegebenen Zeit für den reinen Datenübertragungszyklus wird für die Vorbereitung der Abfrage der nächsten Konsole benötigt. Diese Abfragezeit ist so klein gegenüber der Verarbeitungszeit der Tasteninformation, dass sie für die Abschätzung der Systembelastung vernachlässigt werden kann.

Die für die Verarbeitung von Tasteninformation benötigte Zeit schwankt ganz beträchtlich. Für einen einfachen Tastendruck (keine Antwort) liegt sie im Bereich von 1 - 10 ms.

Für die Verarbeitung einer Auswahlantwort und die Ausgabe der zugehörigen Statistikstanzung werden etwa 75ms benötigt. Für die Aufbereitung, die Bewertung und die Ausgabe der Statistikstanzung bei einer numerischen Antwort sind dagegen total etwa 220ms erforderlich. Wenn angenommen wird, dass alle drei Tasteninformationen etwa gleich häufig vorkommen, so ergibt sich für die mittlere Bedienrate bei der Verarbeitung von Tasteninformation :

$$b_T = \frac{1}{100\text{ms}} = 10/\text{s} \quad (5.5)$$

Für die Abschätzung der mittleren Anforderungsrate kann man von Erfahrungswerten für die durchschnittliche Fragebeantwortungszeit ausgehen. Diese betrug im Wintersemester 1976/77 bei etwas mehr als 900 Lehrprogrammdurchläufen ca. 77,5s pro Antwort. Da jedoch nur etwa die Hälfte aller Tastendrücke Antworten auf Fragen sind, ergibt sich eine mittlere Adressatenreaktionszeit t_A von etwa 40s. Für unsere Rechnung wollen wir jedoch sehr schnell reagierende Adressaten mit $t_A = 20\text{s}$ annehmen.

Für die mittlere Anforderungsrate für die Verarbeitung von Tasteninformation ergibt sich dann bei $k = 32$ angeschlossenen Konsolen :

$$a_T = \frac{k}{t_A} = \frac{32}{20\text{s}} = 1,6/\text{s} \quad (5.6)$$

Aus (5.5) und (5.6) ergibt sich für die Verkehrsintensität bei der Verarbeitung von Tasteninformation :

$$\psi_T = \frac{a_T}{b_T} = 0,16 \quad (5.7)$$

Die Wahrscheinlichkeit P_T , dass bei der Verarbeitung eines Tastendruckes eine Warteschlange mit $n \leq 5$ auftritt, die innerhalb von höchstens 0,5s abgearbeitet werden kann, beträgt somit nach Formel (5.4):

$$P_{T_{n \leq 5}} = 1 - (0,16)^6 = 0,999983 \quad (5.8)$$

Da die einzelnen Tastendrücke statistisch unabhängige Ereignisse sind, ergibt sich für die Wahrscheinlichkeit P_L , dass bei der Bearbeitung eines ganzen Lehrprogramms (ca. 50 Tastendrücke) immer Systemreaktionszeiten $\leq 0,5\text{s}$ auftreten :

$$P_{L_{n \leq 5}} = \left(P_{T_{n \leq 5}} \right)^{50} = 0,9992 \quad (5.9)$$

Es ist also nur bei etwa 0,08% aller durchgearbeiteten Lehrprogramme gelegentlich mit einer Systemreaktionszeit von mehr als 0,5s zu rechnen.

Diese Abschätzung zeigt, dass das PLANETH-Systemprogramm ohne weiteres die Tasteninformation von 32 Lehrkonsolen so verarbeiten kann, dass praktisch keine Wartezeiten auf Computerbedienung auftreten.

5.2.3 Warteschlangenprobleme beim Ausdrucken der Lehrprogramm- Schlussprotokolle

Etwas schwieriger sind die Verhältnisse beim Ausdrucken der Schlussprotokolle am Schluss jedes Lehrprogrammdurchlaufs. Die Vorbereitungen für die Ausgabe einer solchen Meldung dauern etwa 70ms. Wegen der niedrigen Druckgeschwindigkeit der verwendeten Fernschreibmaschine (siehe Kap. 4.2.1) dauert das Ausdrucken eines solchen Schlussprotokolls ca. 21,5s. Damit ein störungsfreier Betrieb des Systems möglich ist, muss also das Ausdrucken aller Meldungen auf der Fernschreibmaschine im Interrupt-Betrieb erfolgen.

Die mittlere Bedienrate für das Ausdrucken der Schlussprotokolle beträgt somit :

$$b_S = \frac{1}{21,5s} = 0,0465/s = 2,791/Min \quad (5.10)$$

Für die Abschätzung der mittleren Anforderungsrate a_S kann man von Erfahrungswerten für den durchschnittlichen Zeitbedarf t_L für die Bearbeitung eines Lehrprogramms ausgehen. Im Wintersemester 1976/77 betrug t_L etwa 38 Minuten pro Lehrprogramm. Um eine gewisse Sicherheitsreserve zu erhalten, sollen für die Rechnung etwas schneller arbeitende Adressaten mit $t_L = 25$ Minuten berücksichtigt werden.

Die mittlere Anforderungsrate a_S für das Ausdrucken von Schlussprotokollen beträgt dann :

$$a_S = \frac{k}{t_L} = \begin{cases} 0,4/Min & \text{bei 10 Konsolen} \\ 1,28/Min & \text{bei 32 Konsolen} \end{cases} \quad (5.11)$$

Aus (5.10) und (5.11) ergibt sich die Verkehrsintensität für das Ausdrucken von Schlussprotokollen :

$$\psi_S = \frac{a_S}{b_S} = \begin{cases} 0,1433 & \text{bei 10 Konsolen} \\ 0,4586 & \text{bei 32 Konsolen} \end{cases} \quad (5.12)$$

Die mittlere Länge \bar{n}_S der entstehenden Warteschlange beträgt nach (5.2) :

$$\bar{n}_S = \begin{cases} 0,167 & \text{für } \psi_S = 0,1433 \quad (k = 10 \text{ Konsolen}) \\ 0,847 & \text{für } \psi_S = 0,4586 \quad (k = 32 \text{ Konsolen}) \end{cases} \quad (5.13)$$

Das Unterprogramm, das das Ausdrucken der Meldungen im Interruptbetrieb steuert, muss einen Pufferbereich für die Speicherung der in der Warteschlange befindlichen Meldungen besitzen. Wegen der beschränkten Grösse des Kernspeichers kann dieser Bereich jedoch nicht beliebig gross gemacht werden. Das heute verwendete PLANETH-Systemprogramm verfügt über 4 Warteschlangenspeicherplätze für Schlussprotokollmeldungen. Wenn die Warteschlange länger wird, führt dies zu einem äusserst unerwünschten Unterbruch des Lehrbetriebs, bis sich ihre Länge wieder auf 4 verkürzt hat. Ein solcher Unterbruch kann zwischen einigen Sekunden und einigen 10 Sekunden dauern. Warteschlangen mit $n > 4$ müssen also unbedingt vermieden werden.

Nach Formel (5.3) ergeben sich für das Auftreten von Warteschlangen mit $n > 4$ beim Ausdrucken der Schlussprotokolle folgende Wahrscheinlichkeiten :

$$P_{S_{n>4}} = \begin{cases} 0,00006 & \text{für } \psi_S = 0,1433 \text{ (k = 10 Konsolen)} \\ 0,0203 & \text{für } \psi_S = 0,4586 \text{ (k = 32 Konsolen)} \end{cases} \quad (5.14)$$

Bei 10 angeschlossenen Konsolen ist also die Wahrscheinlichkeit von Warteschlangenlängen >4 mit 0,006% noch vernachlässigbar klein (heute im Betrieb : 6 Konsolen). Bei 32 Konsolen ist sie jedoch mit über 2% ziemlich gross. Man müsste für diesen Fall durch eine kleine Programm-erweiterung mehr zulässige Warteschlangenplätze schaffen (z.B. 12 statt 4 Plätze) oder ein schneller druckendes Gerät (Liniendrucker) zur Reduktion der Verkehrsintensität ψ_S einsetzen.

Das PLANETH-Systemprogramm ist also in der Lage, die Schlussprotokolle ohne Beeinträchtigung des Lehrbetriebs auszudrucken, wenn im Verhältnis zur Zahl der angeschlossenen Konsolen und zur Geschwindigkeit des Protokolldruckers genügend Warteschlangenspeicherplätze vorhanden sind. Das System reagiert bei der Verarbeitung von Lehrprogrammabschlüssen mindestens ebenso rasch wie bei normalen Antworten (siehe Kap. 5.2.2). Der Adressat erhält wieder innert höchstens $\frac{1}{2}$ s eine Rückmeldung an seiner Konsole (Rücklauf der Konsolenperipherieeinheiten, Lampe ZUM ASSISTENTEN). Bis das zugehörige Schlussprotokoll fertig ausgedruckt worden ist, können wegen der entstehenden Warteschlange vom Programmabschluss gerechnet auch im normalen Betrieb zwischen 20 und 90 s vergehen. Diese Verzögerung ist nach dem Abschluss der Programmbearbeitung jedoch ohne weiteres tragbar. Der Raum, in dem die Protokolle ausgedruckt werden, muss aus Lärmschutzgründen ohnehin vom Lehrmaschinenraum getrennt sein.

5.3 Die Konsolen-Status-Parameter

Wie schon früher erwähnt, benötigt das Systemprogramm für die Durchführung des Lehrbetriebs Informationen über den Betriebszustand jeder Konsole und über das bisherige Lernverhalten des daran arbeitenden Adressaten, damit eine adaptive Steuerung des Lehrprozesses möglich ist. Für eine möglichst genaue Bestimmung des Adressatenzustandes ist neben der Untersuchung der letzten Antwort auch die Berücksichtigung weiter zurückliegender Verhaltensdaten erforderlich [5].

Diese Angaben sind in den Konsolen-Status-Parametern enthalten. Fig. 5.3 zeigt den Aufbau des Konsolen-Status-Parameterspeichers einer Konsole. Er umfasst 13 Worte zu 16 Bits. Zur besseren Ausnützung des Speicherplatzes sind in einzelnen Worten mehrere Parameter gespeichert. Eine kurze Besprechung der gespeicherten Grössen erleichtert das Verständnis für die Möglichkeiten und die Funktion des Systemprogramms.

In den einzelnen Worten sind die folgenden Grössen gespeichert :

- LEPAD : - Anfangsadresse des auf der Konsole gerade bearbeiteten Lehrprogramms.
- Codephase. Gibt an, ob die Konsole unbenützt ist, ob gerade die ETH-Nummer oder die Lehrprogrammnummer eingegeben wird oder ob bereits das eigentliche Lehrprogramm im Gang ist.
- STAND : - Stand des Lehrprogrammzählers (Ort im Lehrprogramm).
- MAZAN : - Maximale Vorgabezeit für die Bearbeitung des aktuellen Lehrschritts bis zur automatischen Programmfortsetzung (in 10-Sek.-Einheiten).
- Totale Anzahl bewerteter Antworten (für Berechnung von Durchschnittspunktzahlen).
- DAUER : - Totale bisherige Dauer des Programmdurchlaufs in Minuten.
- Anzahl der Zusatzzeiten, die der Adressat bei der Bearbeitung des gerade aktuellen Lehrschritts schon verlangt hat.
- ANP : - Summe der bisher erzielten Antwortpunkte für ganz oder teilweise richtig beantwortete Fragen.
- TL (Tastenlampenbild). Gibt an, ob gerade eine Frage mit numerischer Konstruktivantwort, eine Frage mit Auswahlantworten oder eine Kommentarsequenz bearbeitet wird oder ob die Antworttasten vom Computer her gesperrt sind (vergl. Fig. 3.11).

- ZEP : - Summe der bisher erzielten Zeitpunkte. Mass für die Geschwindigkeit der Programmbearbeitung. Je schneller eine Frage im Verhältnis zur Vorgabezeit beantwortet wird, desto höher ist die erzielte Zeitpunktzahl, gleichgültig ob die Antwort richtig oder falsch ist.
- Zähler für Anzahl Antworten auf gleiche Frage. Dient zur Punktereduktion für richtige Antworten im 2. , 3. und 4. Versuch und zur Verhinderung von ewigen Programmschleifen bei ständigen Falschantworten.
- ETHN1 : - Linke 4 Stellen } der 8-stelligen
ETHN2 : - Rechte 4 Stellen } ETH-Nummer des Adressaten.
- LEFRA : - Programmzählerstand der letzten Antwort. Erlaubt die Erkennung von Wiederholungen und die Nachführung des entsprechenden Zählers in ZEP.
- Zähler für Anzahl eingegebene Stellen bei der Eingabe von numerischen Antworten.
 - Zähler für Anzahl Kommastellen bei der Eingabe von numerischen Antworten.
- EVEZ : - Betrag des Exponenten einer eingegebenen numerischen Antwort.
- Vorzeichen dieses Exponenten.
 - Zähler für die bisher verbrauchte Zeit zur Bearbeitung der gerade aktuellen Frage oder Kommentarsequenz (in 10-Sek.-Einheiten).
- NUMER : - Mantisse einer eingegebenen numerischen Antwort (Betrag ≤ 10000 , negative Zahlen = Zweierkomplement).
- TOPAS : - Nummer der zuletzt angesteuerten Tonbandsequenz (für Verarbeitung spezieller BILD-Befehle, siehe Kap. 6).
- Zähler für Anzahl Statistikstanzungen. Verhindert das Auftreten von allzu langen Lehrprogrammdurchläufen, die von den Statistik-Auswerteprogrammen (siehe Kap. 7) nicht mehr verarbeitet werden können.
- SURAD : - Rückkehradresse für den Rücksprung aus Unterprogrammen des Lehrprogramms.

Aufbau des Konsolen-Status-Parameter-Speichers

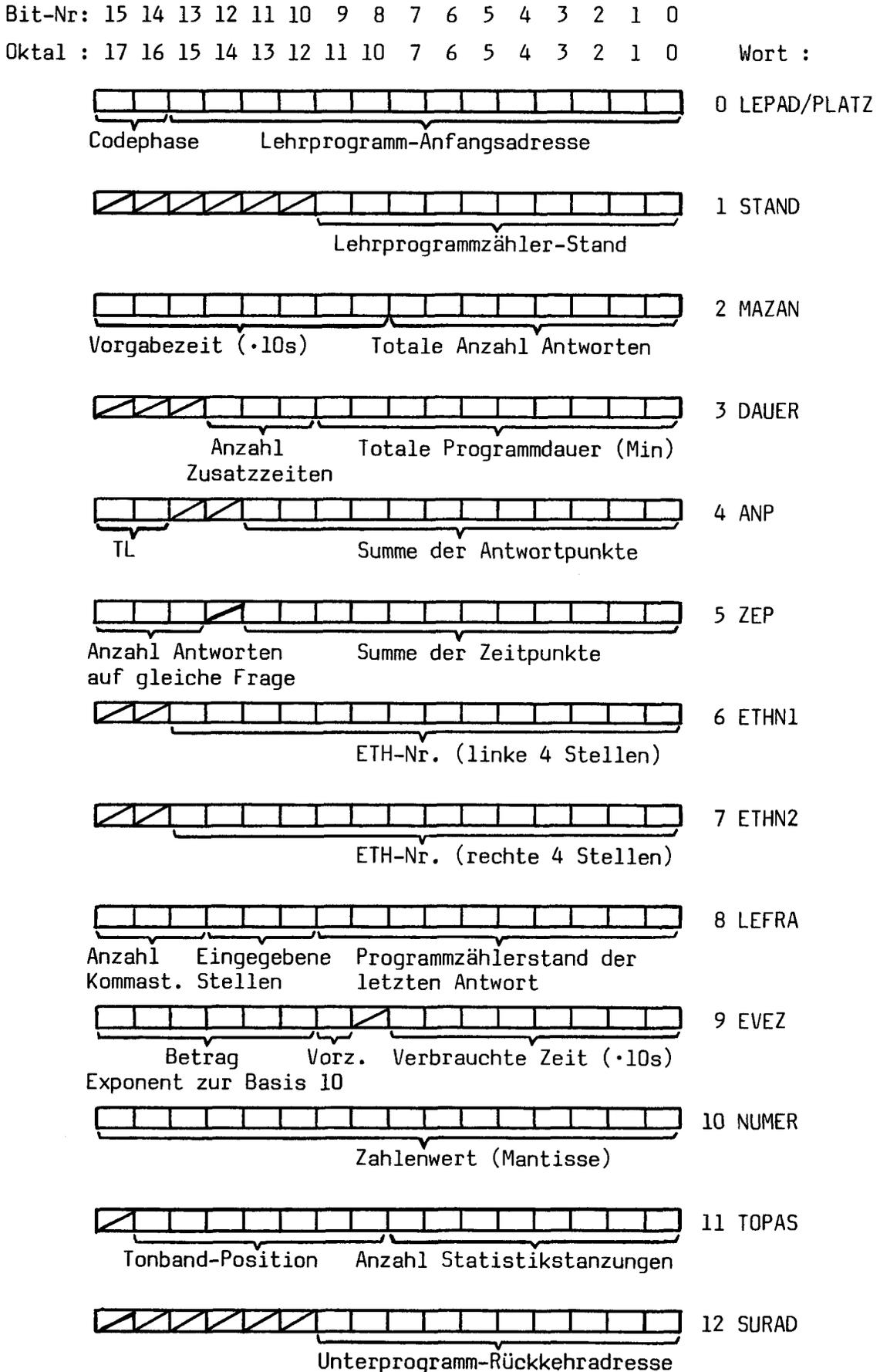


Fig. 5.3 : Aufbau des Konsolen-Status-Parameter-Speichers

5.4 Das PLANETH-System-Hauptprogramm LEMAS

Dieses Programm steuert und koordiniert das Zusammenwirken der vielen Unterprogramme des PLANETH-Systemprogramms. Das wichtige Unterprogramm LEBEV wird in Kap. 5.5 behandelt.

5.4.1 Wichtigste Aufgaben des Hauptprogramms LEMAS

1. Einlesen von Datum und Zeit zu Beginn des Lehrbetriebs
2. Verwaltung des LehrprogrammSpeichers
 - 2.1. Einlesen und Testen der benötigten Lehrprogramme zu Beginn des Lehrbetriebs.
 - 2.2. Einlesen und Testen von zusätzlichen Lehrprogrammen während des Lehrbetriebs (ohne Störung desselben).
 - 2.3. Löschen einzelner nicht mehr benötigter Lehrprogramme während des Lehrbetriebs (ohne Störung desselben).
 - 2.4. Kontrolle, ob immer mindestens ein Lehrprogramm im Speicher.
3. Hauptschleife : Zyklisches Anwählen der einzelnen Konsolen
 - 3.1. Einlesen der Tasteninformation von der Konsole mit Hilfe des Unterprogramms LEDER.
 - 3.2. Wenn STOP-Taste, Systemstop oder Programmende: Lehrprogrammdurchlauf abschliessen.
 - 3.3. Wenn noch kein Lehrprogramm im Gang: Anfangsdaten (ETH-Nummer, Lehrprogrammnummer) einlesen.
 - 3.4. Wenn ZUSATZZEIT-Taste gedrückt: Falls erlaubt, mehr Zeit für die Bearbeitung des Lehrschritts geben.
 - 3.5. Noch verbleibende Restzeit für die Lehrschrittbearbeitung berechnen und anzeigen.

Diese Restzeit berechnet sich nach der Formel :

$$t_{\text{Rest}} = t_{\text{Vor}} + \frac{1}{2}A_{\text{ZZ}} \cdot t_{\text{Vor}} - t_{\text{Verbr}} \quad (5.15)$$

wobei t_{Rest} = Restzeit

t_{Vor} = Vorgabezeit (in MAZAN)

t_{Verbr} = Bisher verbrauchte Zeit

A_{ZZ} = Anzahl verlangte Zusatzzeiten

- 3.6. Wenn Restzeit abgelaufen : Sprung ins Unterprogramm LEBEV, das die Lehrprogrammbeefehle ausführt.
 - 3.7. Wenn WEITER-Taste gedrückt : Sprung ins Unterprogramm LEBEV.
 - 3.8. Wenn eine Frage mit Auswahlantworten beantwortet wurde : Sprung ins Unterprogramm LEBEV.
 - 3.9. Bei Fragen mit numerischer Konstruktivantwort :
Mit Unterprogramm ZAHLD aus den bisher eingetippten Ziffern Zahl definieren und sie mit dem Unterprogramm ZAHLA laufend auf der Konsole anzeigen (Abschluss der Zahleneingabe erfolgt durch Drücken der WEITER-Taste).
 - 3.10. Bei unzulässigen Tastendrücken: Lampe FEHLMANIPULATION anzünden.
 - 3.11. Wenn nötig, neue Anzeigelampen anzünden.
 - 3.12. Gerade aktive Tastenlampen anzünden.
 - 3.13. Nummer der betrachteten Konsole (NRKON) um 1 erhöhen (modulo 32).
 - 3.14. Zeittakt nachführen (Zeittakt = Anzahl Abfragezyklen über alle Konsolen, dient u.a. als Zufallsgrösse).
 - 3.15. Periodische Ausgabeversuche für das Unterprogramm QI04, das die Warteschlange mit den Meldungen für die Fernschreibmaschine abarbeitet.
 - 3.16. Periodischer Test, ob am Schalterregister das Einlesen neuer Lehrprogramme gewünscht wird.
 - 3.17. Periodischer Test, ob am Schalterregister das Löschen eines Lehrprogramms gewünscht wird.
4. Abschluss eines Lehrprogrammdurchlaufs
- 4.1. Beide Projektoren und Tonbandgerät auf 0 setzen, Lampe ZUM ASSISTENTEN anzünden.
 - 4.2. Wenn Schlussprotokoll in der Ausgabe-Warteschlange nicht mehr Platz hat: Warteschlange abarbeiten, bis wieder genug Platz vorhanden.
 - 4.3. Parameter des Schlussprotokolls berechnen und in Warteschlange einfügen.
 - 4.4. Schluss des Lehrprogrammdurchlaufs auf Statistiklochstreifen markieren.

4.5. Abschalten der Konsole veranlassen.

5. Beginn eines Lehrprogrammdurchlaufs: Eingabe der Anfangsdaten

5.1. Nach Drücken der START-Taste : Dia 1 auf rechtem Projektor zeigen. Festhalten, dass Definition der ETH-Nr. folgt.

5.2. ETH-Nummer definieren :
Letzte eingegebene Ziffer mit Unterprogramm ZAHLD an bereits eingegebenen Teil der ETH-Nummer anfügen und neue Zahl mit Unterprogramm ZAHLA auf der Konsole anzeigen.

5.3. Wenn ETH-Nummer fertig eingegeben (nach WEITER) :
Test, ob eingegebene ETH-Nummer zulässig. Wenn ja, festhalten, dass Definition der Lehrprogramm-Nummer folgt.

5.4. Lehrprogramm-Nummer definieren :
Letzte eingegebene Ziffer mit Unterprogramm ZAHLD an bereits eingegebenen Teil der Lehrprogramm-Nr. anfügen und neue Zahl mit ZAHLA auf der Konsole anzeigen.

5.5. Wenn Lehrprogramm-Nummer fertig eingegeben (nach WEITER):
Test, ob gewünschtes Lehrprogramm im Speicher.

Nach dem Auffinden des gewünschten Lehrprogramms :

5.6. Anfang eines neuen Programmdurchlaufs auf dem Statistiklochstreifen markieren.

5.7. Anfangsadresse des Lehrprogramms in LEPAD (siehe Kap.5.3) speichern.

5.8. Lehrprogrammzähler in STAND auf 0 setzen und Sprung ins Unterprogramm LEBEV zur Ausführung der ersten Lehrprogrammbeefehle.

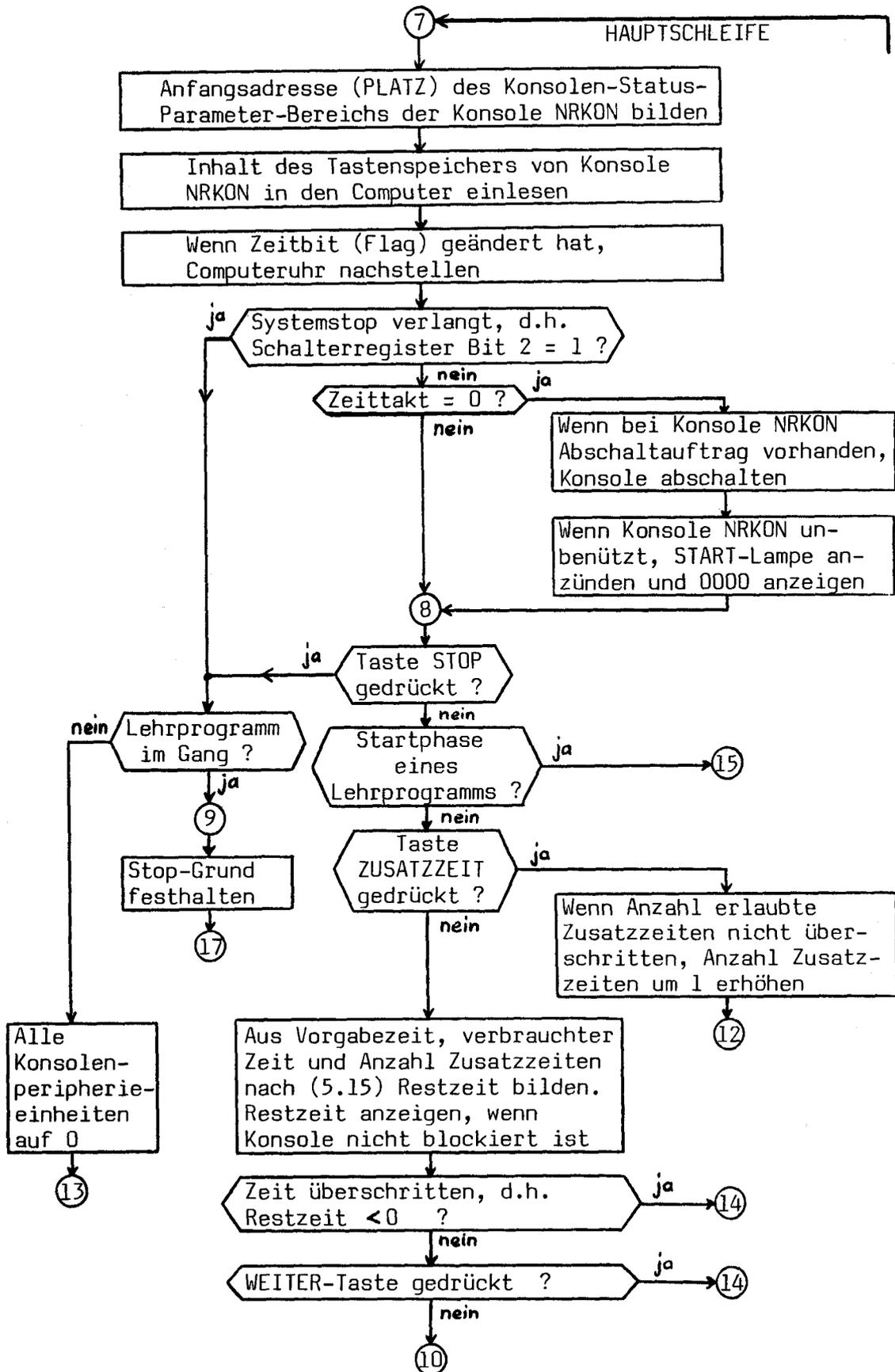


Fig. 5.4b : Flussdiagramm des PLANETH-System-Hauptprogramms (LEMAS)

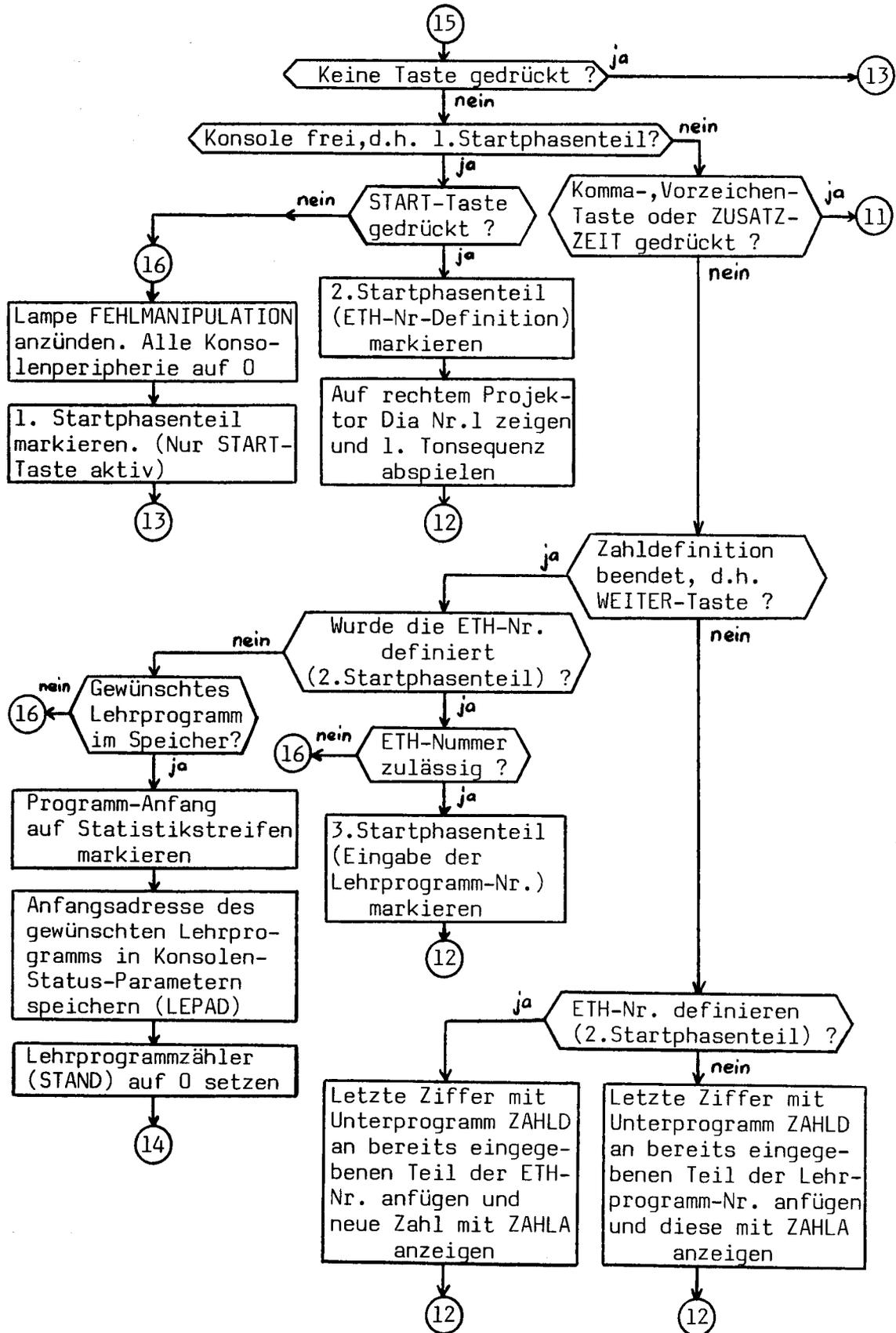


Fig. 5.4d : Flussdiagramm des PLANETH-System-Hauptprogramms (LEMAS)

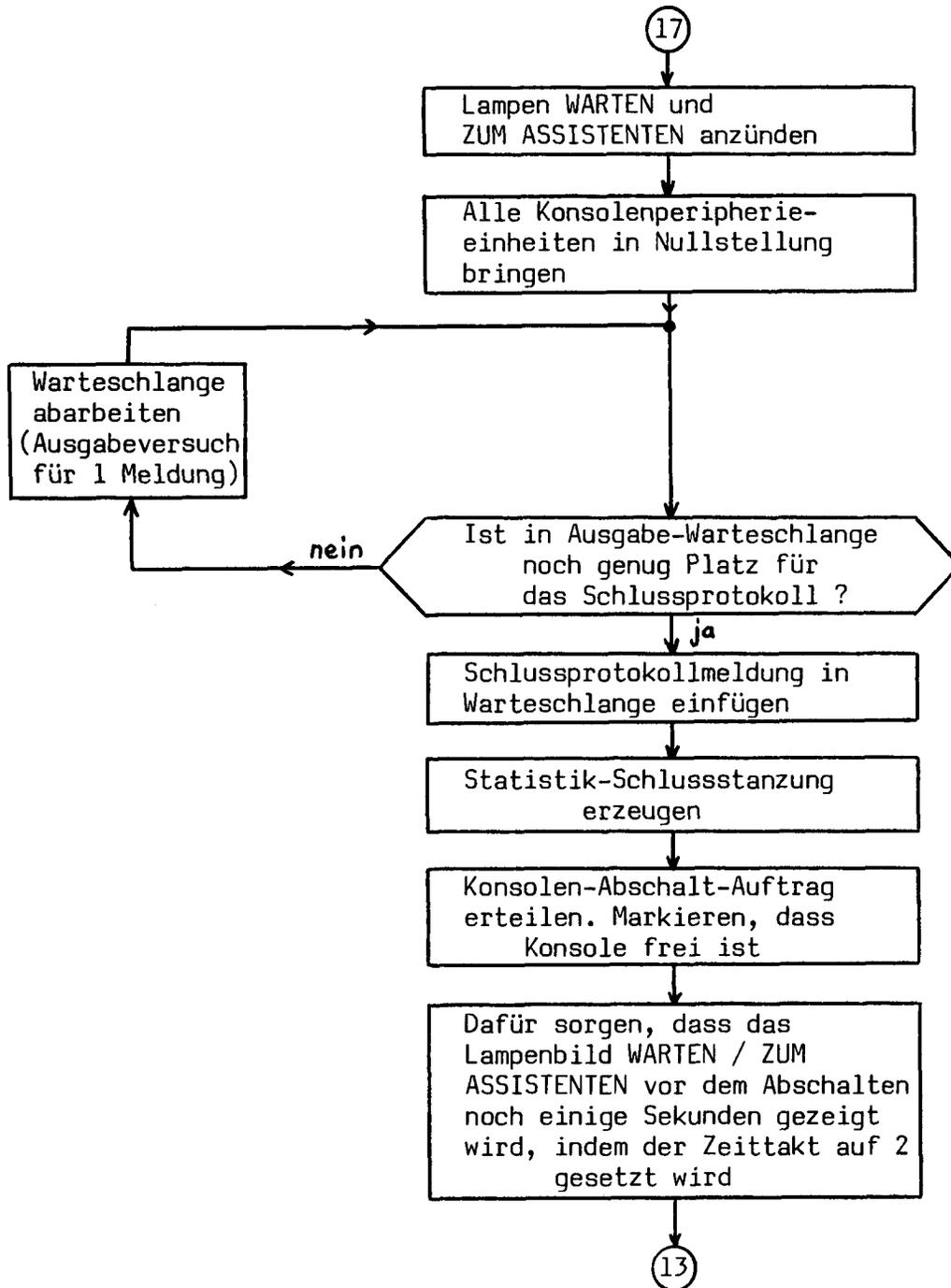


Fig. 5.4e : Flussdiagramm des PLANETH-System-Hauptprogramms (LEMAS)

5.5 Das Unterprogramm LEBEV (Lehrprogramm-Befehls-Verarbeitung)

Dieses Unterprogramm verarbeitet die einzelnen Lehrprogrammbefehle und gibt bei Bedarf mit Hilfe des Unterprogramms LEDER die für die Lehrprozesssteuerung nötigen Daten an die angewählte Lehrkonsole aus.

Jeder Lehrprogrammbefehl besteht aus 2 Computerworten zu 16 Bits, total also 32 Bits. Davon werden jeweils 5 Bits für die Unterscheidung der verschiedenen Lehrprogrammbefehle benötigt. In den übrigen 27 Bits sind die Befehlsparameter enthalten. Weitere Einzelheiten über die Lehrprogrammbefehle sind in Kap. 6.6 zu finden.

Fig. 5.5 zeigt das Flussdiagramm von LEBEV. Nach dem Aufruf des Unterprogramms wird zunächst die Variable BUSY gelöscht. Sie gibt an, ob das Lehrsystem eine Reaktion des Adressaten erwartet oder nicht. Anschliessend wird aus der Lehrprogramm-Anfangsadresse (in LEPAD) und dem Lehrprogrammzählerstand (in STAND) die Adresse des nächsten Lehrprogrammbefehls berechnet. Darauf werden die beiden Worte dieses Befehls aus dem Speicher in die beiden Rechenregister geladen. Nachdem die Art des Befehls festgestellt worden ist, erfolgt ein Sprung in das entsprechende interne Unterprogramm, das diesen Befehl verarbeitet. Diese internen Unterprogramme werden hier nicht im Detail beschrieben, ihre Aufgaben und ihre Funktion sind aus dem Flussdiagramm von Fig. 5.5 ersichtlich.

Ein Lehrprogrammbefehl, der eine Reaktion des Adressaten verlangt (Beantwortung von Fragen), setzt BUSY auf 1. Nach der Ausführung des Befehls wird der Lehrprogrammzähler, der die Position im Lehrprogramm angibt, um 1 erhöht (Ausnahme: Sprungbefehle). Wenn bereits mehr als 32 Lehrprogrammbefehle unmittelbar hintereinander verarbeitet worden sind, so ist das Lehrprogramm höchstwahrscheinlich infolge eines Programmierfehlers in eine ewige Schleife geraten. In diesem Fall erfolgt ein Rücksprung auf (12) im Hauptprogramm LEMAS. Dies geschieht auch, wenn $BUSY > 0$ ist, d.h. wenn eine Reaktion des Adressaten erforderlich ist. Andernfalls wird anschliessend gleich der nächste Lehrprogrammbefehl gesucht und ausgeführt.

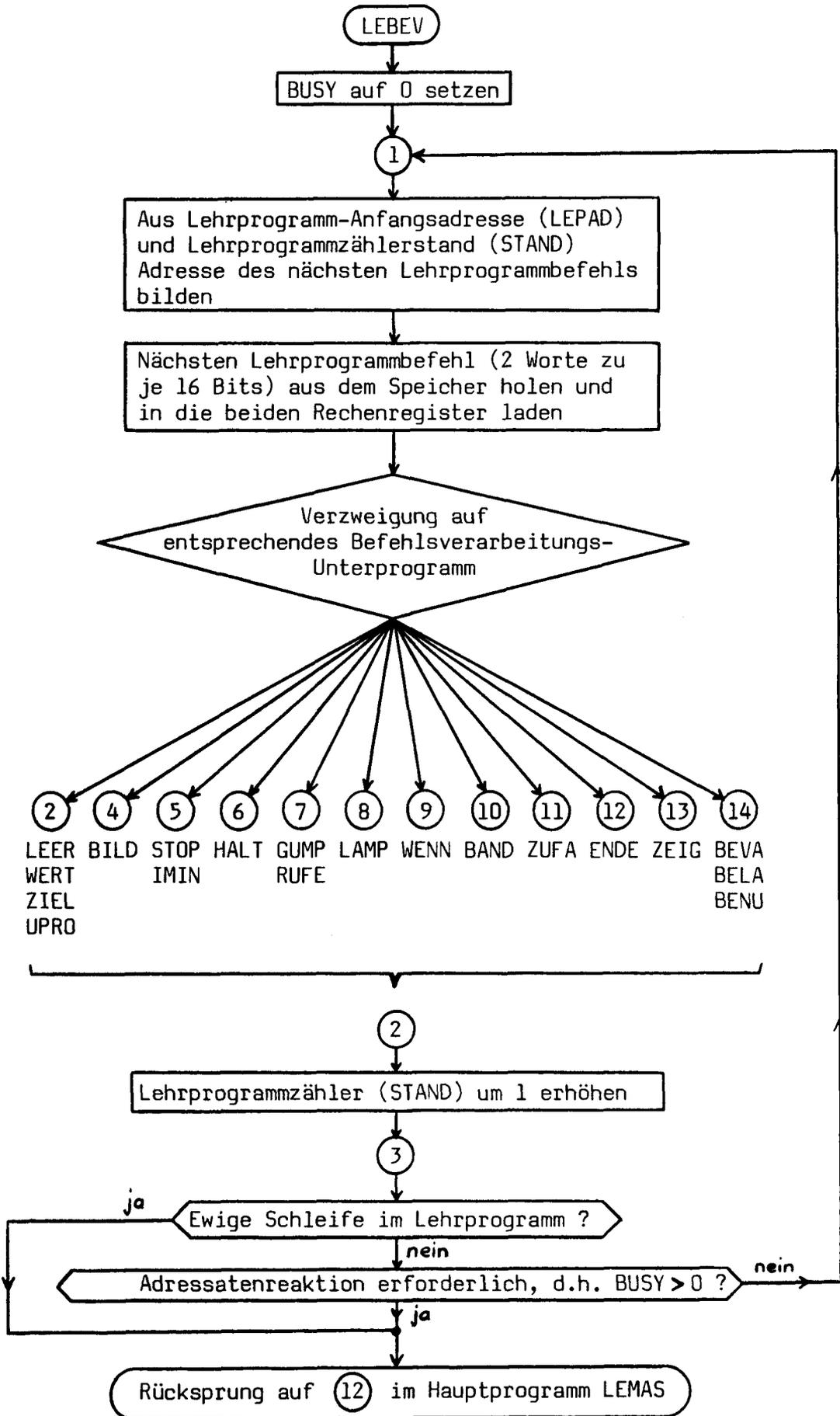


Fig. 5.5a : Flussdiagramm des Unterprogramms LEBEV

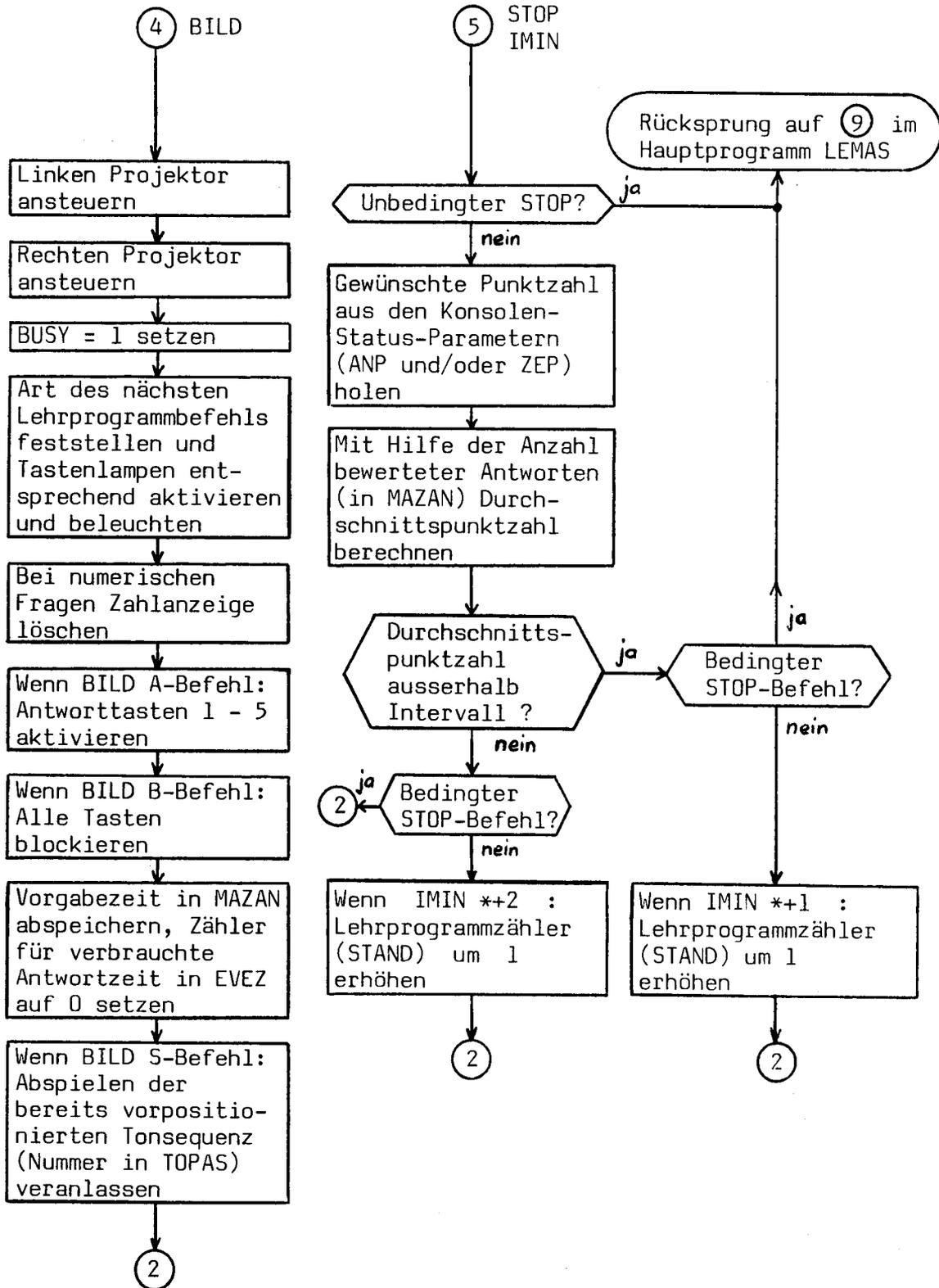


Fig. 5.5b : Flussdiagramm des Unterprogramms LEBEV

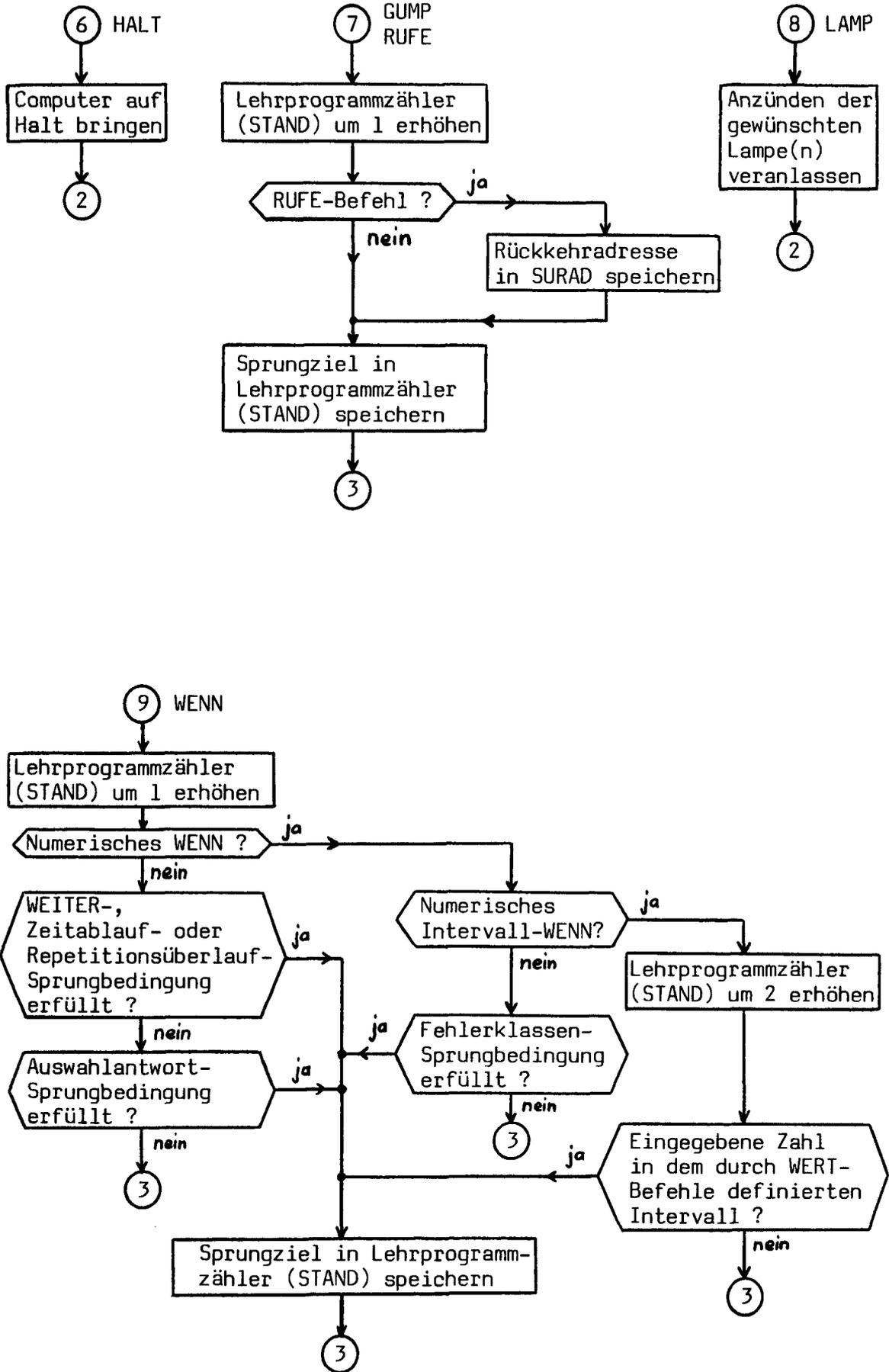


Fig. 5.5c : Flussdiagramm des Unterprogramms LEBEV

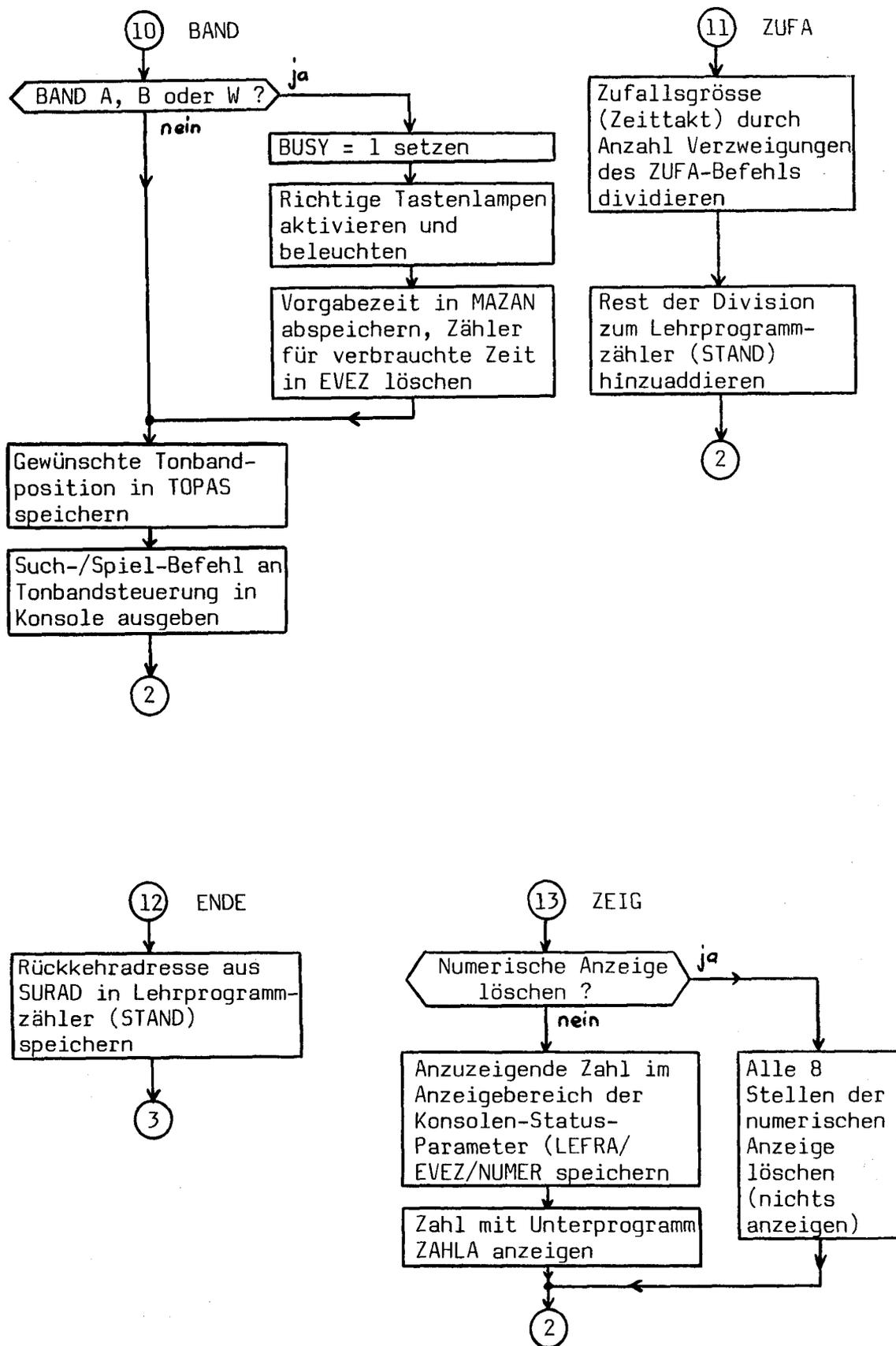


Fig. 5.5d : Flussdiagramm des Unterprogramms LEBEV

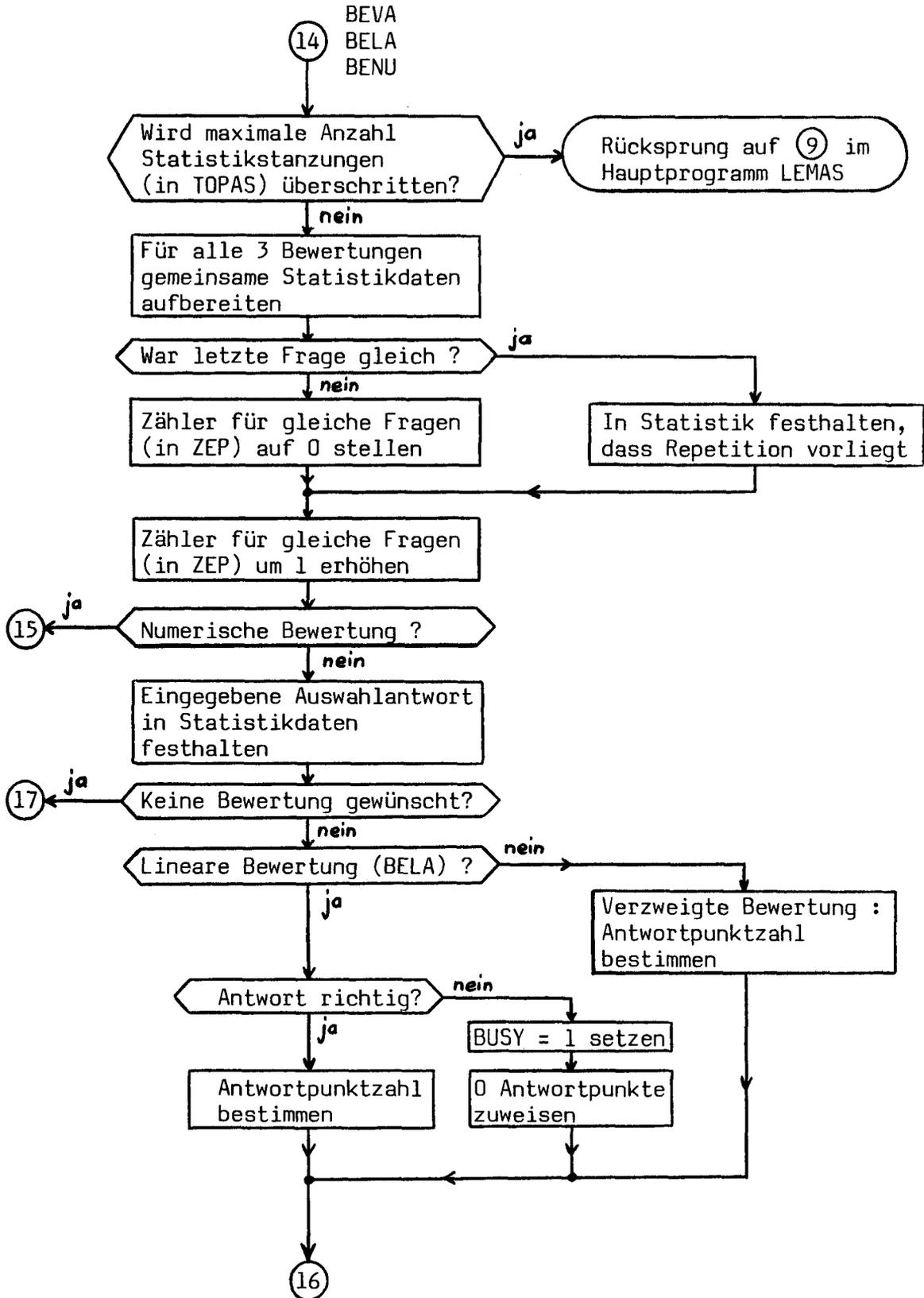


Fig. 5.5e : Flussdiagramm des Unterprogramms LEBEV

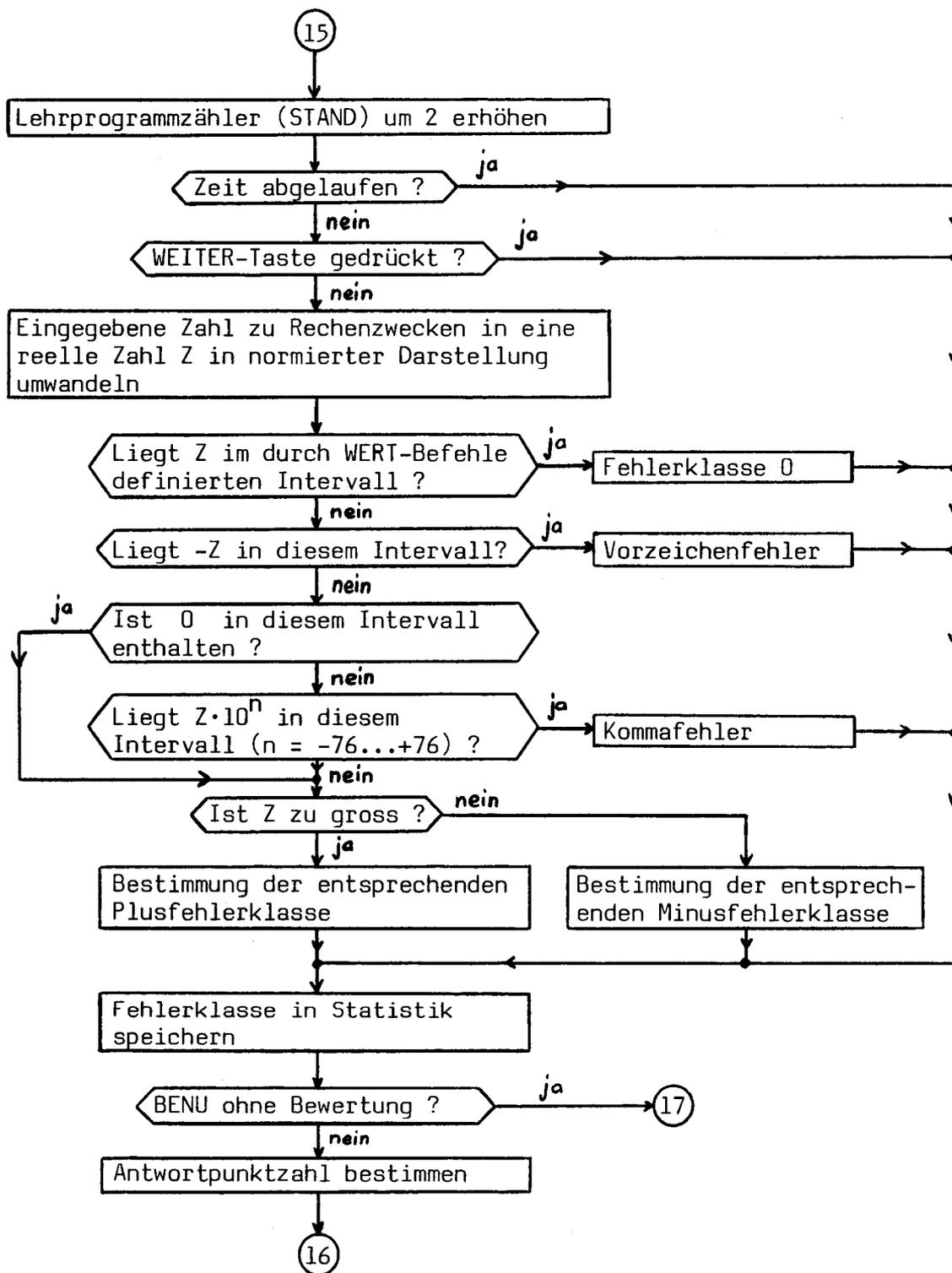


Fig. 5.5f : Flussdiagramm des Unterprogramms LEBEV

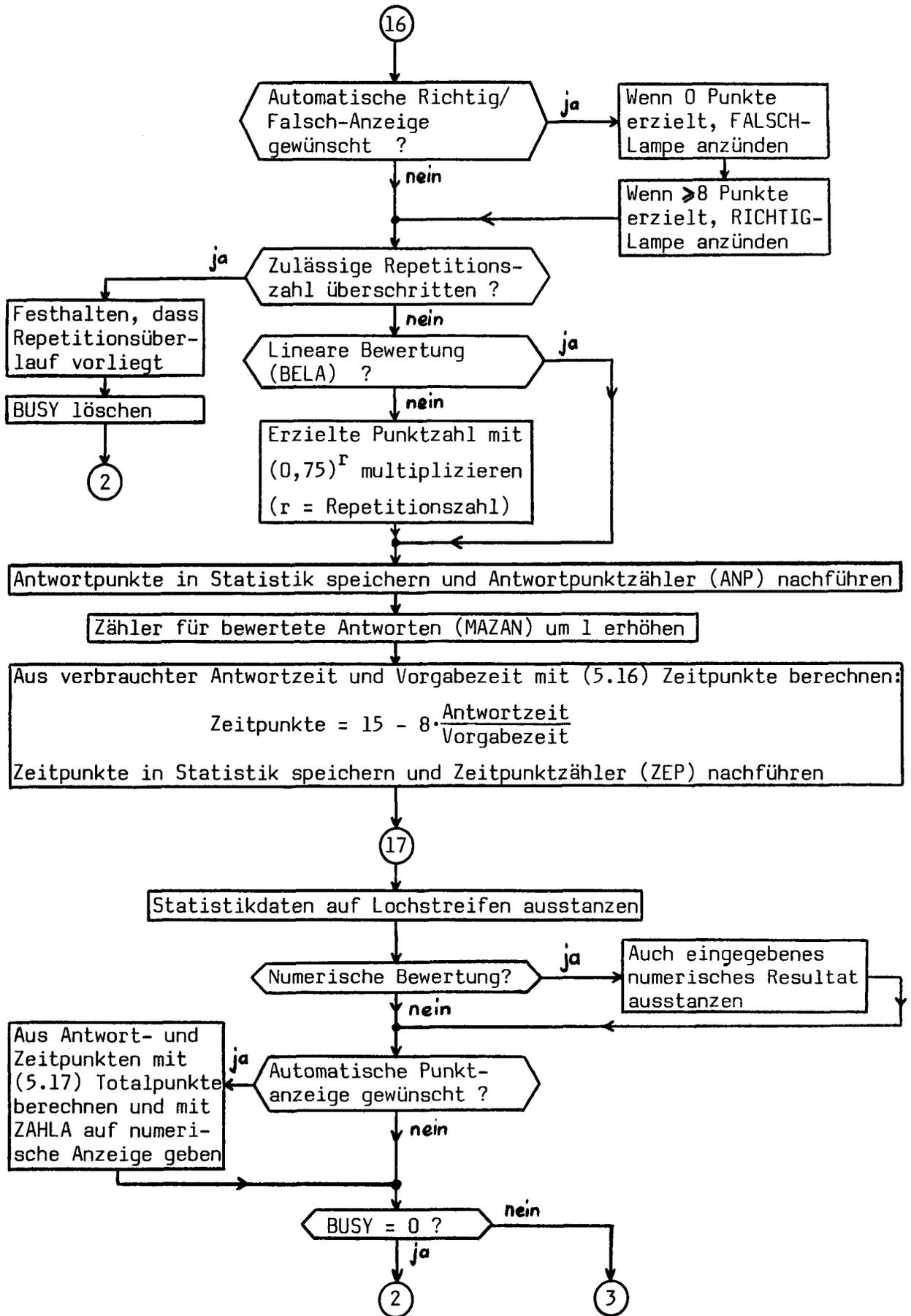


Fig. 5.5g : Flussdiagramm des Unterprogramms LEBEV

5.5.1 Die Bewertung der eingegebenen Antworten

Im internen Bewertungs-Unterprogramm, das die Lehrprogrammanweisungen BELA, BEVA und BENU (siehe Kap. 6.6) verarbeitet, wird für jede Antwort auf eine Frage die erzielte Antwortpunktzahl ermittelt. Diese Antwortpunktzahl ist in den Parametern dieser Anweisungen enthalten und liegt zwischen 0 und 15 Punkten.

Damit man auch eine Angabe hat, wie schnell eine Frage im Verhältnis zur ursprünglichen Vorgabezeit beantwortet wird, wird bei jeder Antwort ausser der Antwortpunktzahl auch eine Zeitpunktzahl nach folgender Formel berechnet :

$$z = 15 - 8 \frac{t_{\text{Verbr}}}{t_{\text{Vor}}} \quad (\geq 0) \quad (5.16)$$

wobei z = Zeitpunktzahl
 t_{Verbr} = Verbrauchte Antwortzeit
 t_{Vor} = Vorgabezeit (in MAZAN)

Wenn der Adressat also sehr schnell antwortet, erhält er 15 Zeitpunkte, wenn er dagegen erst nach der doppelten Vorgabezeit reagiert, 0 Zeitpunkte. Die erzielten Zeitpunkte hängen jedoch nicht davon ab, ob die Antwort richtig oder falsch war.

Alle Antwortpunkte und Zeitpunkte werden in einem Antwortpunktzähler (in ANP) und in einem Zeitpunktzähler (in ZEP) laufend aufsummiert, damit man jederzeit einen Ueberblick über das ganze bisherige Lernverhalten des Adressaten hat.

Für eine möglichst einfache Bewertung der Gesamtleistung (z.B. für eine Notengebung) ist es oft erwünscht, eine einzige Punktzahl zur Verfügung zu haben, die sowohl die gegebenen Antworten als auch die benötigte Zeit berücksichtigt. Zu diesem Zweck wird auch noch eine sogenannte Totalpunktzahl P_T nach folgender Formel berechnet :

$$P_T = P \left(1 + \frac{Z}{a} \cdot \frac{p_5}{200} \right) \quad (5.17)$$

wobei P_T = Totalpunktzahl
 P = Summe der bisher erzielten Antwortpunkte
 Z = Summe der bisher erzielten Zeitpunkte
 a = Anzahl bisher gegebene Antworten
 p_5 = Zeitgewicht für die Berechnung der Totalpunktzahl.
Steuerparameter, der in der PROG-Anweisung (siehe Kap. 6.6.2.1) definiert wird ($0 \leq p_5 \leq 7$).

Wenn ein Adressat sehr langsam arbeitet, wird $Z = 0$ und die Totalpunktzahl gleich der Antwortpunktzahl. Dies gilt natürlich auch für $p_5 = 0$. Arbeitet er dagegen schneller, wird $Z/a > 0$ und kann im Maximum den Wert 15 erreichen ($0 \leq Z/a \leq 15$). Wenn $p_5 > 0$ ist, wird die Totalpunktzahl P_T dann grösser als die Summe der Antwortpunkte P . Mit $p_5 = 1$ und $Z/a = 15$ ist P_T beispielsweise um 7,5% grösser als P .

Schnelles Arbeiten wird aber nur honoriert, wenn auch richtige Antworten gegeben worden sind. Falls $P = 0$ ist, trägt der Ausdruck $Z \cdot p_5 / 200a$ zur Totalpunktzahl nichts bei.

Bei jedem Bewertungsbefehl wird die bisher erzielte Totalpunktzahl auf die numerische Anzeige der Lehrkonsole ausgegeben, wenn dies nicht ausdrücklich vom Lehrprogrammautor durch einen bestimmten Steuerparameter (p_3) in der PROG-Anweisung (siehe Kap. 6.6.2.1) unterdrückt wird. Um deutlich hervorzuheben, dass die Totalpunktzahl und nicht bloss irgend eine andere Zahl angezeigt wird, erscheinen auf den drei Stellen am linken Rand des Anzeigefeldes drei Punkte.

5.6 Die Aufgaben der übrigen Unterprogramme

In diesem Kapitel werden kurz die übrigen Unterprogramme besprochen, die gewisse wichtige Teilaufgaben übernehmen.

5.6.1 Das Unterprogramm LEDER : Datenverkehr mit Party-Line

Dieses Unterprogramm dient zur Ein- und Ausgabe von Party-Line-Daten. Für die Ausgabe von Daten an die 6 Einheiten der gerade angewählten Lehrkonsole stehen 6 Eingänge (ENTRY-POINTS) zur Verfügung :

DIA (linker Projektor) ,	DIB (rechter Projektor) ,
LAMPE (Lampensteuerung) ,	RESTT (Zeitsteuerung) ,
NUMA (Numerische Anzeige),	TON (Tonbandsteuerung) .

Ein weiterer Eingang (INPUT) dient zum Einlesen der Tasteninformation der angewählten Konsole.

In LEMAS und LEBEV müssen jeweils nur die 8 Datenbits aufbereitet werden und der richtige Unterprogramm-Eingang aufgerufen werden, der Rest wird von LEDER automatisch besorgt.

Bei Störungen auf der Party-Line während der Datenausgabe druckt LEDER eine Fehlermeldung auf dem Fernschreiber heraus und veranlasst nach ca. 3 Sekunden selbständig einen weiteren Ausgabeversuch. Erst nach 4 erfolglosen Versuchen wird das Programm angehalten. Uebertragungsfehler beim Einlesen von Tasteninformation werden ebenfalls mit dem Fernschreiber gemeldet, die fehlerhaften Daten werden jedoch einfach ignoriert.

5.6.2 Das Unterprogramm ZAHLD : Definition von Zahlen

Bei der Eingabe von numerischen Konstruktivantworten werden die einzelnen Ziffern nacheinander mit der Konsolentastatur eingegeben. Eine Verarbeitung der eingetippten Zahl darf jedoch erst erfolgen, wenn diese vollständig eingegeben worden ist und der Adressat die Zahleneingabe durch Drücken der WEITER-Taste beendet hat.

Das Unterprogramm ZAHLD bildet aus den einzelnen eingetippten Ziffern laufend den Wert der bisher eingegebenen Zahl in den Konsolenparametern LEFRA, EVEZ und NUMER (siehe Kap. 5.3). Durch das Drücken der START-Taste kann dieser Zahlendefinitionsbereich bei Eingabefehlern gelöscht werden. Mit ZAHLD können sowohl 4-stellige ganze Zahlen (mit Vorzeichen) als auch reelle Zahlen mit 4-stelliger Mantisse und 2-stelligem Exponenten, die betragsmässig im Bereich 10^{-38} bis 10^{+38} (oder bei 0) liegen, verarbeitet werden.

5.6.3 Das Unterprogramm ZAHLA : Anzeige von Zahlen

Das Unterprogramm ZAHLA ermöglicht die Anzeige von Zahlen im Zahlendefinitionsbereich auf der zugehörigen Konsole. Damit kann bei der Eingabe numerischer Konstruktivantworten die bisher eingetippte Zahl laufend angezeigt werden. Mit ZAHLA können 4-stellige ganze Zahlen (mit Vorzeichen) und reelle Zahlen mit 4-stelliger Mantisse und 2-stelligem Exponenten angezeigt werden. ZAHLA wird auch für die Anzeige der folgenden Zahlen eingesetzt :

- ETH-Nummer während Startphase
- Lehrprogrammnummer während Startphase
- Bisher erzielte Punktzahl in den Bewertungsbefehlen
- Von Lehrprogramm mit ZEIG-Befehl verlangte Zahlen .

5.6.4 Die Unterprogramme für die Textausgabe auf der Fernschreibmaschine : QIO4, .TTY. und LETEX

Zu Beginn der Entwicklung des PLANETH-Systems wurden die Schlussprotokolle und Fehlermeldungen mit Hilfe von Fortran-Unterprogrammen ausgedruckt. Diese Unterprogramme belegten mit den zugehörigen Hilfsunterprogrammen (Formatierprogramm und Ein-/Ausgabeprogramm) vom Computerhersteller ca. 3000 Worte, also etwa 37% des verfügbaren Kernspeichers. Dies war natürlich auf die Dauer nicht tragbar, denn für das Ausdrucken der PLANETH-Meldungen wurden jeweils nur kleine Teile der entsprechenden Hilfsunterprogramme effektiv benützt. Diese Programme enthielten überdies die für die Ausgabe von Fernschreibermeldungen unbedingt nötigen Warteschlangenplätze (siehe Kap. 5.2.3) noch nicht.

Deshalb wurden eigene Unterprogramme für die Ausgabe von Fernschreibermeldungen entwickelt, die auch Warteschlangenplätze für insgesamt 4 komplette Schlussprotokolle enthalten und zusammen nur etwa 1100 Worte belegen. Dabei konnten weitgehend von meinen Kollegen H. Müller (.TTY.) und B. von Arx (QIO4) entwickelte Unterprogramme verwendet werden, die für die Verwendung im PLANETH-System allerdings noch etwas modifiziert werden mussten. .TTY. enthält das eigentliche Interrupt-Ein- und Ausgabeprogramm für den Betrieb der Fernschreibmaschine. QIO4 ist für das Management der Warteschlange zuständig. Der Text der Meldungen und eine einfache Formatangabe für jede von ihnen ist in LETEX gespeichert.

Das Unterprogramm QIO4 besitzt 4 Eingänge :

- OUTPT bewirkt den Eintrag einer Meldung in die Ausgangs-Warteschlange.
- OUTQ initialisiert das Ausdrucken der nächsten in der Warteschlange befindlichen Meldung mit Hilfe von .TTY. , wenn die letzte Meldung beendet ist.
- STOUT stellt die Anzahl der in der Warteschlange noch vorhandenen freien Plätze fest.
- CLTTY druckt alle in der Warteschlange befindlichen Meldungen aus, bevor eine Rückkehr ins Hauptprogramm erfolgt.

Fig. 5.6 zeigt einige von diesen Unterprogrammen ausgedruckte Fernschreibermeldungen. Oben stehen die beim Start des PLANETH-Systemprogramms ausgegebenen Meldungen (Abfrage des Datums und der Zeit, Quittungen für eingegebene Lehrprogramme, eventuelle Fehlerbeanstandungen). Unten folgen zwei Schlussprotokolle nach Abschluss eines Lehrprogrammdurchlaufs.

LEHRSYSTEM PLANETH

DATUM/ZEIT (TAG,MONAT,JAHR,STUNDEN,MINUTEN) : 9.6.78 8.12

GELADENE LEHRPROGRAMME :

1111

1112

1113

1114

1115

1121

1122

1123

1124

1125

2121

2122

LEHRSYSTEM LAEUFT

ETH-ZUERICH , PLANETH(1) , 9.6.78 , 8.55 UHR
STUDENT : 77901247
LEHRPROGRAMM: 1125
PUNKTZAHL : 68 (61/105/9A)
DURCHSCHNITT: 7.55 PUNKTE PRO ANTWORT
DAUER (MIN) : 36
PROGRAMMENDE

ETH-ZUERICH , PLANETH(3) , 9.6.78 , 8.57 UHR
STUDENT : 75903450
LEHRPROGRAMM: 2121
PUNKTZAHL : 49 (49/60/5A)
DURCHSCHNITT: 9.80 PUNKTE PRO ANTWORT
DAUER (MIN) : 28
STOP-TASTE

Fig. 5.6 : Einige Fernschreibermeldungen des PLANETH-Systemprogramms

5.6.5 Das Unterprogramm BIMAP :

Kompaktspeicherung von Daten und grösser-kleiner Vergleiche

Dieses Unterprogramm führt gewisse elementare Funktionen aus, die im PLANETH-System häufig gebraucht werden, die jedoch im Satz der Maschineninstruktionen des hp 2116 B nicht enthalten sind. Teilweise sind diese Funktionen bei neueren Computern als Maschineninstruktionen verfügbar.

BIMAP besitzt 3 Eingänge :

- BIMA ermöglicht sowohl beim Laden als auch beim Speichern den direkten Zugriff zu einzelnen Bits oder Bitgruppen innerhalb eines Computerwortes. Die Information, welche Bits gemeint sind, ist in einem Steuerwort unmittelbar nach dem Aufruf des Unterprogramms enthalten. Die Adresse des angewählten Computerwortes ist entweder in einem Rechenregister (B-Register) oder in einem 2. Steuerwort enthalten. BIMA wird hauptsächlich zur kompakten Speicherung der Konsolen-Status-Parameter verwendet (siehe Kap. 5.3).
- ALTB } erlauben grösser-kleiner Vergleiche
- AGTB } zwischen den beiden Rechenregistern.

6. DIE HERSTELLUNG DER LEHRPROGRAMME

6.1 Probleme beim Einsatz von Lehrmaschinenprogrammen

Das wohl wichtigste Hindernis beim Einsatz von Lehrmaschinen im Unterricht ist der meist hohe zeitliche und materielle Aufwand für die Entwicklung der zugehörigen Lehrprogramme. Die dadurch anfallenden hohen Festkosten müssen auf eine möglichst grosse Anzahl Adressaten verteilt werden können (vergl. Kap. 1.4), damit die Verwendung von Lehrmaschinen rentabel wird.

Andererseits soll ein Lehrprogramm natürlich möglichst gut auf die Lehrveranstaltung zugeschnitten sein, in der es eingesetzt wird, damit sein Wirkungsgrad optimal ist. Auf der Hochschulstufe wird es deshalb am besten vom betreffenden Dozenten selbst oder von einem seiner Assistenten geschrieben. Da die Verbreitung eines solchen Lehrprogramms meist nicht über die eigene Hochschule hinausgehen wird, muss es einfach und billig herzustellen sein, und wenn der vermittelte Stoff sich ändert, sollen Programmanpassungen leicht fallen. Damit möglichst geringe Kosten resultieren, soll die Herstellung mit den Mitteln eines Hochschulinstituts ohne Beizug von spezialisiertem Fachpersonal (z.B. Grafiker) erfolgen können. Dabei muss natürlich auf übertriebenen Perfektionismus verzichtet werden. In den meisten Fällen genügen einfache farbige Handzeichnungen an Stelle von aufwendigen Grafiken vollauf.

Auf Mittel- und Volksschulstufe ist die Zahl der potentiellen Adressaten eines Lehrprogramms wesentlich grösser als auf Hochschulstufe. Infolge des föderalistisch aufgebauten Schulsystems ist aber der Einsatz des gleichen Lehrprogramms an vielen verschiedenen Orten zumindest in der Schweiz eher unwahrscheinlich. Auch auf dieser Stufe ist also die Möglichkeit enorm wichtig, Lehrprogramme mit einfachsten Mitteln und geringsten Kosten selbst herstellen und abändern zu können.

In der Industrie verfügen straff organisierte Grossbetriebe ebenfalls über grosse potentielle Adressatenzahlen. Wenn die betriebsinterne Schulung zentral organisiert ist, können unter Umständen auch Lehrmaschinen eingesetzt werden, die einen grösseren Aufwand für die Lehrprogrammherstellung erfordern und bei denen Programmänderungen

nur mit sehr grossen Kosten durchzuführen sind, wie das BASF-Lehrgerät LG 5100 [14]. Nach den Erfahrungen bei der Herstellung von gegen 20 Lehrprogrammen über Elektrotechnik am Mikrowellenlabor der ETH ist es aber praktisch unmöglich, ein Programm von Anfang an völlig fehlerfrei zu machen. Besonders in stark verzweigten Lehrprogrammen zeigen sich manche Fehler erst nach vielen Programmdurchläufen. Auch beim Einsatz von Lehrmaschinen in der Industrie ist deshalb eine leichte Programmänderungsmöglichkeit sehr wünschenswert.

6.2 Die Lehrstoffspeicher beim Lehrsystem PLANETH

Als Lehrstoffträger werden beim PLANETH-Lehrsystem normale 24mm·36mm Kleinbilddiapositive und Kompakt-Tonband-Kassetten verwendet. Sie sind überall erhältlich, billig und millionenfach bewährt. Der Umgang mit Fotoapparat und Tonbandgerät ist heute wohl jedem geläufig, der irgendwie pädagogisch tätig ist. Nachträgliche Korrekturen sind sehr einfach: Man muss nur die betreffenden Dias ersetzen und falsche Tonsequenzen neu aufnehmen. Die in Kap. 6.1 postulierte Möglichkeit, Programme leicht selbst herzustellen und abzuändern, ist hier also sicher gegeben. Details über die Dia- und Tonbandherstellung befinden sich in Kap. 6.7 und 6.8 .

6.3 Die für die Lehrprogrammerstellung nötigen Arbeiten

Bei der Lehrprogrammerstellung werden zuerst einige didaktisch-pädagogische Vorarbeiten durchgeführt. Anschliessend erfolgt die technische Realisierung. Die didaktischen Vorarbeiten umfassen in einer ersten Phase die Lehrziel-fixierung, die Adressatenanalyse, die Stoffsammlung und die Kurszusammenstellung. Nach Abschluss dieser ersten Phase wird der Lehrstoff meist in Form des sogenannten Basaltextes [13] fixiert. Bei reinen Übungsprogrammen ist eine Basaltextherstellung nicht unbedingt notwendig. Oft genügt eine Zusammenstellung der vorgesehenen Fragen und der dazu gehörenden Erklärungen und Kommentare. All diese Arbeiten sind noch medienunabhängig, weshalb der dafür notwendige Aufwand bei allen Lehrmaschinen gleich ist.

In einer zweiten Phase wird der Lehrstoff didaktisch aufbereitet. Der Basaltext wird dabei in die einzelnen Lehrschritte zergliedert. Es werden Bild- und Textentwürfe hergestellt und eventuell ein Flussdiagramm des Lehrprogramms gezeichnet. Bei diesen Arbeiten sollen bereits die technischen Möglichkeiten der verwendeten Lehrmaschine berücksichtigt werden.

Für die technische Realisierung von PLANETH-Lehrprogrammen werden zunächst von allen Bild- und Textvorlagen Reinzeichnungen (meist farbige Handzeichnungen) bzw. Reinschriften (mit Schreibmaschine) erstellt. Anschliessend werden die Bilder und die schriftlich zu präsentierenden Texte fotografiert. Die gesprochenen Texte werden zusammen mit eventuellen Musiksequenzen und den nötigen Markierimpulsen (siehe Kap. 3.11.1) auf einem qualitativ hochwertigen Spulentonbandgerät mit 19cm/s Bandgeschwindigkeit aufgenommen. Darauf werden die entwickelten Dias eingerahmt und in die 80-plätziges Rundmagazine eingefüllt. Der Ton wird vom Mutterband auf Kassetten (mit 4,75cm/s Bandgeschwindigkeit) kopiert. Anschliessend erfolgt die Codierung des Lehrprogramms im engeren Sinn, das die vom steuernden Computer benötigten Informationen über Struktur und Ablauf des ganzen Lehrprogramms enthält (vergl. Kap. 5). Schliesslich wird das fertige Lehrprogramm vom Autor auf der Lehrkonsole ausgetestet.

Für die didaktischen Vorarbeiten der zweiten Phase und teilweise auch für die technische Realisierung können nach Frank [13] sogenannte Formaldidaktiken eingesetzt werden, die den Zeitaufwand gegenüber der direkten Programmherstellung zum Teil stark reduzieren. Diese Formaldidaktiken sind Computerprogramme, die aus denjenigen Teilen des Basaltextes, welche die wesentliche Information enthalten (Basalwörter), gemäss mehr oder weniger starren Regeln ein Lehrprogramm erzeugen. Formaldidaktikprogramme laufen jedoch nur auf Grosscomputern und schränken die didaktische Freiheit wesentlich ein (z.B. Beschränkung auf Skinner-Lehrprogramme oder ständige Wiederholung fast gleich aufgebauter stereotyper Sätze). Ueberdies können sie nur von speziell geschulten Didaktikern benützt werden. Am vorteilhaftesten ist ihr Einsatz für die Erzeugung von Lehrprogrammen, die nur aus Text bestehen.

Für viele Fachrichtungen (z.B. die hier speziell interessierende Ingenieurausbildung) werden jedoch Lehrprogramme benötigt, die neben Text viel Information in Form von Schemas, Diagrammen, Zeichnungen und Bildern enthalten, die nicht direkt von einem Computer erzeugt werden können. Der Einsatz eines Grosscomputers für die Programmherstellung würde zudem gegen das wichtige Prinzip des Lehrsystems PLANETH verstossen, dass für Erzeugung, Betrieb und statistische Auswertung der Lehrprogramme der gleiche Kleincomputer verwendet werden soll. Die Notwendigkeit, für die Programmherstellung einen externen Grosscomputer beizuziehen, würde diese stark komplizieren und viele

potentielle Lehrprogrammautoren von vorneherein abschrecken. Aus diesen Gründen wurde beim PLANETH-System auf den Einsatz des Computers für die Aufbereitung des Lehrstoffs (Bilder und Text) verzichtet.

Bei PLANETH-Lehrprogrammen werden Text und Bilder also direkt vom Lehrprogrammautor erzeugt. Er kann natürlich für manche Arbeiten Hilfskräfte (Sekretärinnen, Studenten) beiziehen. Die Reinschrift von Textvorlagen wird beispielsweise am Mikrowellenlabor in der Regel von der Institutssekretärin erstellt. Die anschließende Herstellung der Dias und Tonkassetten ist leicht durchzuführen und stellt keine besonderen Probleme. Durch die Entwicklung einer einfachen Autorensprache und eines dazu gehörenden Uebersetzungsprogramms wurde versucht, den Aufwand für die Codierung der Lehrprogramme möglichst gering zu halten. Bei andern Lehrmaschinen [14, 15, 16] ist die Erzeugung des Lehrprogrammcodes oft recht mühsam und muss auf speziellen Codierblättern vorgenommen werden, die nur schwer lesbar sind. Zudem werden dazu meist mehrere Arbeitsgänge benötigt. Die Autorensprache SESAM zum Lehrsystem PLANETH ist dagegen relativ leicht verständlich (Einzelheiten siehe Kap. 6.6). Sie umfasst gegenwärtig 20 verschiedene Befehle. Das Uebersetzungsprogramm wandelt in einem einzigen Arbeitsgang die in SESAM geschriebenen Lehrprogrammbeefehle in einen für das Lehrsystem PLANETH verständlichen speziellen Binärcode um (Details siehe Kap. 6.5).

6.4 Der Aufwand für die Programmherstellung in Funktion der Komplexität des Lehrprogrammes beim Lehrsystem PLANETH

Ein besonderer Vorteil des Lehrsystems PLANETH besteht darin, dass sich Lehrprogramme mit verschiedenen Komplexitätsgraden und entsprechend verschiedenem Aufwand realisieren lassen.

Ganz einfache Lehrprogramme bieten den ganzen Lehrstoff auf Dias an und verwenden nur Fragen mit Auswahlantworten. Für die Codierung eines solchen Programms werden nur wenige SESAM-Befehle benötigt. Der zeitliche Aufwand für die Programmherstellung ist somit ziemlich klein. Er liegt erfahrungsgemäss etwa zwischen 80 und 140 Arbeitsstunden pro Lehrprogrammstunde.

Etwas kompliziertere Lehrprogramme sind ähnlich aufgebaut, sie verwenden jedoch neben Fragen mit Auswahlantworten auch Fragen mit numerischen Konstruktivantworten. Dadurch wird das Programm viel interessanter und lehrreicher. Der Aufwand für die Herstellung eines solchen Programms nimmt nur geringfügig zu. Er beträgt etwa 100 - 160 Stunden pro Lehrprogrammstunde.

Ganz raffiniert aufgebaute Lehrprogramme nützen dagegen alle technischen Möglichkeiten des Lehrsystems PLANETH aus. Sie präsentieren den Lehrstoff mit Bild und Ton und verwenden Fragen mit Auswahlantworten und numerischen Antworten. Solche Programme besitzen viele Verzweigungen, die nicht nur von der letzten Antwort, sondern auch von der bisher erbrachten Gesamtleistung oder von der Arbeitsgeschwindigkeit abhängen. Zur Erhöhung der Sicherheit vor Mogelversuchen werden oft auch Zufallsverzweigungen eingesetzt. Der Aufwand für die Programmherstellung steigt dementsprechend auf etwa 150 - 250 Stunden pro Lehrprogrammstunde.

Die oben angegebenen Zeiten für die Herstellung von Lehrprogrammen gelten nur dann, wenn zu Beginn keine brauchbaren Unterlagen über den Lehrstoff vorliegen. Sind jedoch solche Unterlagen vorhanden (z.B. Lehrprogramm in Buchform für Instruktionsprogramm, erprobte Fragen für Übungsprogramm), so ist der Aufwand für die Programmherstellung natürlich noch wesentlich geringer.

Im praktischen Lehrbetrieb wollen oft mehrere Adressaten gleichzeitig auf verschiedenen Lehrkonsolen das gleiche Lehrprogramm absolvieren. Damit dies möglich ist, müssen zu jedem Lehrprogramm mehrere Sätze von Lehrstoffspeichern hergestellt werden. Dazu werden für das Einrahmen der Dias, Einfüllen der Dias in die Diamagazine, Bespielen der Tonkassette und das Austesten weitere 2 - 4 Stunden pro Satz benötigt. Der grösste Teil der Arbeiten kann jedoch von Hilfskräften (Studenten) ausgeführt werden.

6.5 Das Uebersetzungsprogramm (Compiler) für die Autorensprache SESAM

Dieses Programm, welches die SESAM-Befehle in einen dem PLANETH-Systemprogramm verständlichen binären Code übersetzt, wurde von H.Müller in etwa 1½-jähriger Arbeit geschrieben [17]. Es läuft auf dem gleichen Computer, der normalerweise den Lehrbetrieb steuert.

Die Lehrprogrammbeefehle können entweder interaktiv oder mit Lochkarten eingegeben werden. Jeder eingegebene Befehl wird vom Uebersetzungsprogramm sofort auf syntaktische Fehler geprüft und nötigenfalls beanstandet. Nach Abschluss der Lehrprogrammeingabe werden verschiedene logische Tests vorgenommen, welche die Erkennung von Fehlern im Programmablauf gestatten. Sämtliche Fehlermeldungen erfolgen in leicht verständlichem Klartext, also nicht bloss durch Angabe einer Fehler-

nummer. Bei syntaktischen Fehlern wird überdies das erste beanstandete Zeichen genau angegeben. Ein fehlerfreies Programm kann anschliessend in binärer Form auf einen Lochstreifen ausgestanzt werden, der später vom PLANETH-Systemprogramm eingelesen werden kann.

Bei der interaktiven Herstellung von Lehrprogrammen ist das ursprüngliche SESAM-Quellenprogramm nirgends gespeichert. Das Uebersetzungsprogramm kann aber den erzeugten binären Lochstreifen wieder in die ursprünglichen Anweisungen zurückverwandeln. Im Prinzip genügt also der binäre Lochstreifen mit dem Lehrprogrammcode für die Programmarchivierung.

Mit speziellen Steueranweisungen kann das Uebersetzungsprogramm im interaktiven Betrieb auch für Korrekturen und Aenderungen an bereits bestehenden Lehrprogrammen verwendet werden. Es erlaubt das Ersetzen und Löschen unerwünschter Programmbefehle und das Einfügen von neuen Befehlen an beliebiger Stelle. Dabei werden wenn nötig alle Sprungziele automatisch korrigiert.

Die Eingabe von Lehrprogrammen mit Lochkarten hat vor allem den Vorteil, dass die benötigte Computerzeit für die Programmübersetzung sehr stark reduziert wird. Da der für den PLANETH-Betrieb verwendete Computer keinen Lochkartenleser besitzt, erfolgt die Uebersetzung solcher Lehrprogramme mit einem andern, besser mit Peripherieeinheiten ausgerüsteten Kleincomputer. Bei auf Lochkarten geschriebenen Programmen können zudem die einzelnen Lehrschritte ausführlich kommentiert werden, was die Lesbarkeit des Programms wesentlich erhöht.

6.6 Die Autorensprache SESAM zum Lehrsystem PLANETH

6.6.1 Prinzipielle Eigenschaften und Befehlsvorrat

Ein Lehrprogramm besteht aus einer Folge von Programmbefehlen oder Anweisungen. Jede dieser Anweisungen steht auf einer Zeile. Die einzelnen Programmbefehle werden vom PLANETH-Systemprogramm einer nach dem andern ausgeführt, bis eine Adressatenreaktion erforderlich ist oder der Programmdurchlauf mit einem STOP-Befehl abgeschlossen wird.

Eine solche Adressatenreaktion erfordern der BILD-Befehl, gewisse Arten von BAND-Befehlen sowie der BELA-Befehl nach einer falschen Antwort. Die Bedeutung dieser Befehle wird später erklärt.

Die einzelnen Zeilen und damit auch die einzelnen Anweisungen werden mit 1 beginnend fortlaufend nummeriert, damit auch Sprünge im Lehrprogramm möglich sind. Die Eingabe dieser Zeilennummern ist fakultativ. Meist ist es bequemer,

beim interaktiven Erstellen eines Programms bei Bedarf eine numerierte Liste der bisher eingegebenen Befehle ausdrucken zu lassen. Es ist auch möglich, einzelne Zeilen mit Marken zu versehen und statt auf die Zeilennummer direkt auf die entsprechende Marke zu springen.

6.6.1.1 Aufbau eines Lehrprogrammbefehls

Jeder Lehrprogrammbefehl besteht aus einem Befehlswort, das aus vier Buchstaben aufgebaut ist, und eventuell aus einigen Parametern, die aus Buchstaben und/oder Zahlen zusammengesetzt sind.

Beispiel : BILD A,17,29,Z90

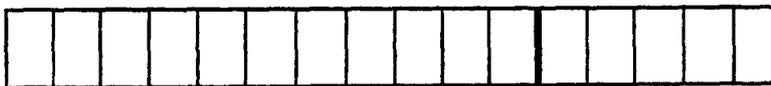
BILD ist das Befehlswort, A, 17, 29 und Z90 sind die Parameter.

Die Gesamtheit aller Parameter einer Programmanweisung wird als Parameterliste bezeichnet. Die Parameterliste muss vom Befehlswort durch mindestens einen Leerschlag und/oder mindestens ein Komma getrennt sein. Die einzelnen Parameter sind voneinander mindestens durch ein Komma zu trennen. Ein Strichpunkt ; bedeutet Ende der Anweisung. Allfällige weitere Zeichen auf der gleichen Zeile werden vom Uebersetzungsprogramm als Kommentar interpretiert und ignoriert.

Das Uebersetzungsprogramm wandelt jede Lehrprogrammanweisung in 2 aufeinanderfolgende Computerworte zu 16 Bits um. In den 5 Bits mit dem geringsten Gewicht (rechts aussen) des 1. Wortes ist die Code-nummer (0 - 31) des Befehls enthalten. Im Rest des 1. Wortes und im 2. Wort sind die zugehörigen Parameter gespeichert.

Bit-Nr.

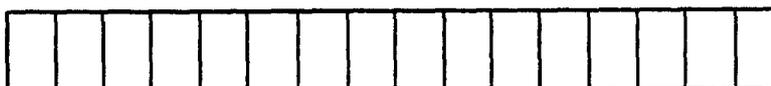
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



1. Wort (Befehlswort)

Bit-Nr.

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



2. Wort
(Parameterwort)

Fig. 6.1 : Prinzip der Codierung eines Lehrprogrammbefehls

6.6.1.2 Uebersicht über die einzelnen Lehrprogrammbeefhle

Die total 20 verschiedenen Programmbefehle lassen sich nach ihrer Funktion in 4 Gruppen einteilen. Die Befehle werden hier nur kurz vorgestellt. Nähere Einzelheiten und insbesondere Angaben über die zugehörigen Parameter finden sich in den folgenden Kapiteln und in [17] und [18].

1. Deklarationsbefehle

PROG Dieser Befehl steht immer am Anfang eines Lehrprogramms. Er dient zur Festlegung der vierstelligen Lehrprogrammnummer und einiger im ganzen Programm gültiger Steuerparameter.

WERT dient zur Definition eines Zahlenwerts, der von andern Befehlen (BENU, numerisches WENN) benötigt wird.

UPRO dient zur Deklaration des Namens (1 oder 2 Buchstaben) eines Unterprogramms im Lehrprogramm. Unterprogramme müssen am Schluss des Lehrprogramms deklariert werden.

ZIEL dient zur Deklaration einer Marke als Sprungziel, die auch im Binär-code erhalten bleibt.

Diese vier Deklarationsbefehle sind nicht ausführbar. Sie werden vom Systemprogramm bei der Ausführung der Lehrprogrammanweisungen übersprungen.

2. Befehle zur Ansteuerung der Konsolenperipherieeinheiten

BILD Dieser Befehl veranlasst die beiden Projektoren, je ein Dia anzuwählen und während einer bestimmten Zeit zu zeigen. Nach jedem BILD-Befehl wird anschliessend die Abarbeitung der Lehrprogrammbeefhle unterbrochen, da eine Reaktion des Adressaten erwartet wird. Erfolgt innerhalb der Vorgabezeit keine solche Reaktion, wird das Programm automatisch fortgesetzt.

BAND Dieser Befehl veranlasst das Aufsuchen und nötigenfalls auch das Abspielen einer bestimmten Tonsequenz durch das eingebaute Kassettentonbandgerät. Auch gewisse Arten von BAND-Befehlen (BAND A, B und W) unterbrechen die weitere Verarbeitung der Programmanweisungen, bis eine Adressatenreaktion oder der Ablauf der Vorgabezeit erfolgt.

LAMP erlaubt das Anzünden bestimmter Anzeigelampen (z.B. FEHLMANIPULATION, ZUM ASSISTENTEN) an der Lehrkonsole.

ZEIG veranlasst die Anzeige einer bestimmten Zahl auf der numerischen Anzeige der Konsole.

3. Befehle für die Antwortbewertung

In einem Lehrprogramm kann man Informationslehrschrirte und Fragelehrschrirte unterscheiden. Informationslehrschrirte dienen zur Präsentation des Lehrstoffs oder zur Kommentierung von früher gegebenen Antworten. Sie erfordern keine Entscheidung des Adressaten und werden durch Drücken der WEITER-Taste oder durch Zeitablauf abgeschlossen. Fragelehrschrirte erfordern dagegen eine Antwort des Adressaten. Im Lehrprogramm erkennt man sie daran, dass der entsprechende BILD-Befehl (oder evtl. BAND A-Befehl) unmittelbar von einem der drei Bewertungsbefehle BELA, BEVA oder BENU gefolgt wird. In der vom Uebersetzungsprogramm ausgedruckten Liste sind sie mit einem Stern * gekennzeichnet.

BELA (Bewertung, linearer Ablauf) ist speziell geeignet für die Bewertung von Auswahlantworten in linearen Lehrprogrammteilen. Wenn der Adressat richtig geantwortet hat, erhält er eine bestimmte Antwortpunktzahl. Darauf wird die nächste Programmanweisung ausgeführt. War die Antwort dagegen falsch, erhält der Adressat die Gelegenheit, nochmals einen Versuch zur Beantwortung der Frage zu machen. Das Programm bleibt deshalb beim BELA-Befehl stehen und wartet auf die nächste Antwort.

BEVA (Bewertung, verzweigter Ablauf) dient zur Bewertung von Auswahlantworten in verzweigten Lehrprogrammstrukturen. Jede der fünf möglichen Antworten kann mit einer bestimmten Antwortpunktzahl bewertet werden, eine starre Unterscheidung in richtige oder falsche Antworten entfällt also. Anschliessend wird immer die nächste Programmanweisung ausgeführt. Mit entsprechenden WENN-Befehlen lassen sich zu jeder Antwort passende Verzweigungen programmieren.

BENU (Bewertung numerischer Antworten) dient zur Bewertung von numerischen Konstruktivantworten. Die BENU-Anweisung ist immer von zwei WERT-Befehlen gefolgt, welche die Grenzen des Intervalls bestimmen, in dem das richtige Resultat liegen muss. BENU erlaubt die individuelle Bewertung von vollständig richtigen Antworten, von knapp daneben liegenden Resultaten, von Kommafehlern und Vorzeichenfehlern. Die Art eines eventuellen Fehlers wird gespeichert. Anschliessend wird auch beim BENU-Befehl die nächste Anweisung ausgeführt. Mit numerischen WENN-Befehlen können Verzweigungen je nach der Art des gemachten Fehlers realisiert werden.

4. Befehle zur Steuerung des Programmablaufs

STOP dient zum Abschluss des Lehrprogrammdurchlaufs. Der STOP-Befehl kann sowohl unbedingte als auch bedingte sein. Ein unbedingter STOP-Befehl wird immer ausgeführt, ein bedingter jedoch nur, wenn eine bestimmte durchschnittliche Punktzahl pro Antwort ausserhalb gewisser angegebener Grenzen liegt.

GUMP ist der unbedingte Sprungbefehl. Es erfolgt ein Sprung auf die im GUMP-Befehl angegebene Zeilennummer oder Marke.

WENN ist der bedingte Sprungbefehl. Er erlaubt, die Ausführung eines Sprunges auf eine bestimmte Zeile oder Marke von einer bestimmten Bedingung abhängig zu machen. Diese Bedingung bezieht sich auf die letzte gegebene Antwort.

Es gibt 3 Kategorien solcher Bedingungen :

a) Auswahlantwort - WENN :

Der Sprung wird ausgeführt, wenn eine der im WENN-Befehl angegebenen Antworttasten betätigt wurde. Auf Wunsch können auch Sprünge programmiert werden, wenn die WEITER-Taste gedrückt, die Zeit abgelaufen oder die zulässige Anzahl Repetitionen überschritten wurde.

b) Numerisches Fehlerklassen - WENN

Der Sprung wird ausgeführt, wenn eine der im WENN-Befehl spezifizierten Fehlerarten (Fehlerklassen) vorliegt.

c) Numerisches Intervall - WENN :

Der Sprung wird ausgeführt, wenn das eingegebene numerische Resultat in einem bestimmten Intervall liegt, das durch die zwei folgenden WERT-Befehle definiert ist.

IMIN ermöglicht Verzweigungen, die nicht nur von der letzten Antwort, sondern vom gesamten bisherigen Verhalten des Adressaten abhängen. Es können Verzweigungen in Funktion der folgenden Grössen programmiert werden :

- a) Durchschnittliche Antwortpunktzahl pro Antwort
- b) Durchschnittliche Zeitpunktzahl pro Antwort
- c) Durchschnittliche Totalpunktzahl pro Antwort

Die Totalpunkte werden aus den Antwort- und Zeitpunkten berechnet. Einzelheiten darüber siehe Kap. 5.5.1 .

ZUFA ist der Befehl für zufällige Verzweigungen.

RUFE dient zum Aufrufen eines Unterprogramms vom Hauptprogramm aus. In Unterprogrammen sind RUFE-Befehle dagegen verboten (Keine Verschachtelung von Unterprogrammen möglich).

ENDE bewirkt die Rückkehr aus einem Unterprogramm ins Hauptprogramm.

LEER ist, wie der Name sagt, ein leerer Befehl und hat keine Wirkung.

HALT hält das PLANETH-Systemprogramm an, wobei im T-Register des Computers der programmierte HALT-Code erscheint. Diese Anweisung darf nur zu Testzwecken verwendet werden. In fertig ausgetesteten Lehrprogrammen ist sie streng verboten.

6.6.2 Die Deklarationsbefehle

6.6.2.1 Der PROG-Befehl

Jedes Lehrprogramm beginnt mit einer PROG-Anweisung. Sie muss und darf nur auf Zeile 1 stehen. Sie dient dazu, dem Programm eine Nummer zuzuordnen und einige globale Steuerparameter, die im ganzen Programm gültig sind, zu definieren.

Der Befehl hat folgende Form : PROG p₁, p₂, p₃, p₄, p₅, p₆

oft auch geschrieben als : PROG p₁ (p₂, p₃, p₄, p₅, p₆)

Die Parameter p₁ - p₆ haben folgende Bedeutung :

p₁ ist die Programmnummer. Sie besteht aus 4 Dezimalziffern (0 - 9) und dient zur Kennzeichnung der einzelnen Lehrprogramme.

p₂ gibt an, ob die RICHTIG- und FALSCH-Lampe an der Konsole automatisch angesteuert werden soll, was die Einsparung vieler LAMP-Befehle erlaubt. Bei eingeschalteter Automatik wird in den Bewertungsbefehlen entschieden, welche Lampe angezündet werden soll.

Beim linearen Bewertungsbefehl BELA ist der Entscheid einfach. Die gegebene Antwort ist entweder richtig oder falsch, so dass einfach die entsprechende Lampe angesteuert werden muss.

In den andern beiden Bewertungsbefehlen BEVA und BENU erfolgt die Entscheidung auf Grund der erzielten Antwortpunktzahl. Bei mehr als 7 Punkten wird die RICHTIG-Lampe, bei 0 Punkten die FALSCH-Lampe angezündet. Die RICHTIG-/FALSCH-Lampenautomatik wird ausge-

schaltet, wenn für p_2 eine Null (0) oder ein Minuszeichen (-) gesetzt wird. Jedes andere Zeichen (z.B. F) schaltet die Automatik ein.

p_3 gibt an, ob bei Bewertungsbefehlen auf der numerischen Anzeige der Lehrkonsole jeweils automatisch die bisher erzielte Totalpunktzahl angezeigt werden soll. Die Totalpunktzahl berechnet sich aus den Antwortpunkten, Zeitpunkten und dem Steuerparameter p_5 nach der Formel (5.17).

Diese automatische Totalpunktanzeige wird unterdrückt, wenn p_3 auf Null (0) oder Minus (-) gesetzt wird. Jedes andere Zeichen (z.B. P) schaltet die Punktanzeigeautomatik ein.

p_4 gibt an, wie oft der Adressat bei einem Lehrschritt Zusatzzeit verlangen kann ($0 \leq p_4 \leq 7$). Jedesmal, wenn die Taste ZUSATZZEIT gedrückt wird, verlängert das PLANETH-Systemprogramm die zur Lehrschrittbearbeitung noch verbleibende Restzeit gemäss Formel (5.15) um die halbe Vorgabezeit, sofern die Anzahl verlangte Zusatzzeiten A_{ZZ} den Wert p_4 noch nicht überschritten hat. Andernfalls wird die Taste ZUSATZZEIT ignoriert.

p_5 gibt das sogenannte Zeitgewicht an, das den Einfluss der Zeitpunktzahl auf die Berechnung der Totalpunkte nach Formel (5.17) bestimmt ($0 \leq p_5 \leq 7$). Nähere Angaben siehe Kap. 5.5.1 .

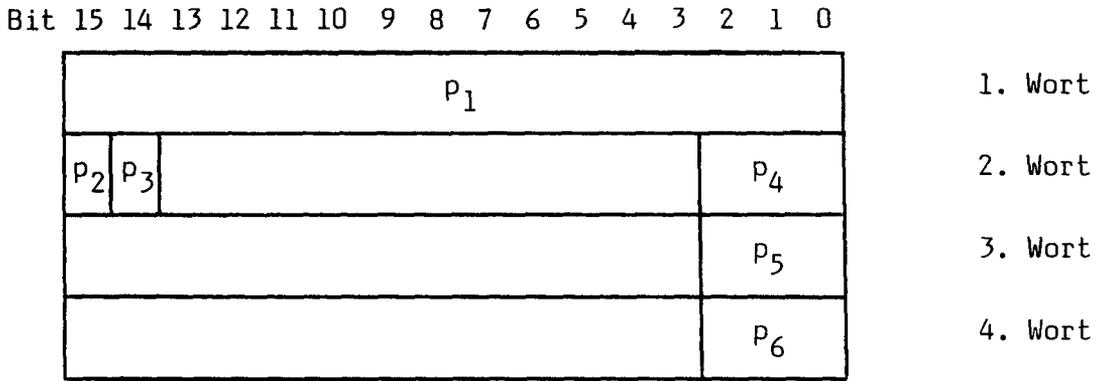
p_6 ist ein Reserveparameter, der vorläufig noch keine Bedeutung hat. Das Uebersetzungsprogramm akzeptiert die Werte $p_6 = 0 \dots 7$.

Beispiel für eine PROG-Anweisung :

PROG 1112 (F,P,3,2,0)

Codierung des PROG-Befehls :

PROG belegt im Gegensatz zu allen andern Befehlen 4 statt nur 2 Worte zu 16 Bits. Fig. 6.2 zeigt die Codierung der einzelnen Steuerparameter.



Legende :

- p₁ : Programmnummer (ganze Zahl im Bereich 1 - 9999), rechtsbündig
- p₂ : $\left\{ \begin{array}{l} \text{Bit 15} = 1, \text{ wenn } p_2 = F, \text{ Lampenautomatik RICHTIG/FALSCH ein} \\ \text{Bit 15} = 0, \text{ wenn } p_2 = 0 \text{ oder } - , \text{ d.h. Lampenautomatik aus} \end{array} \right.$
- p₃ : $\left\{ \begin{array}{l} \text{Bit 14} = 1, \text{ wenn } p_3 = P, \text{ d.h. automatische Totalpunktanzeige} \\ \text{Bit 14} = 0, \text{ wenn } p_3 = 0 \text{ oder } - , \text{ d.h. keine Totalpunktanzeige} \end{array} \right.$
- p₄ : Anzahl erlaubte Zusatzzeiten, Bits 2 - 0. Bereich: 0 - 7.
- p₅ : Zeitgewicht für Totalpunktberechnung, Bits 2 - 0. Bereich: 0 - 7.
- p₆ : Reserveparameter, Bits 2 - 0. Bereich: 0 - 7.

Die von den Steuerparametern p₁ - p₆ nicht belegten Bits dienen als weitere Ausbaureserve.

Fig. 6.2 : Codierung des PROG-Befehls

6.6.2.2 Der WERT-Befehl

Der Wert-Befehl hat genau einen Parameter, nämlich eine reelle Zahl. Diese Zahl wird für die Ausführung anderer Befehle (BENU, numerisches WENN) benötigt. Sie besteht aus einer Mantisse von 1 bis 4 signifikanten Stellen und fakultativ aus einem Exponenten zur Basis 10. Die Mantisse darf einen Dezimalpunkt enthalten. Intern wird sie jedoch als ganze Zahl dargestellt, wobei der Exponent jeweils so verändert wird, dass der Zahlenwert wieder stimmt.

Beispiel:

WERT 1.234 E-8
 Mantisse Exponent zur Basis 10

Interne Darstellung dieses Beispiels :

WERT 1234 E-11

Die betragsmässig grössten und kleinsten darstellbaren Zahlen betragen :

9999 E+34 bzw. 1 E-38

Codierung :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

	EXP	S		0	1	1	0	0
Z (Mantisse der Zahl)								

Legende :

Z = Zahlenwert (Mantisse), ganzzahlig. Negative Zahlen werden als Zweierkomplement dargestellt. Bereich: -9999 ... +9999 .

EXP = Betrag des Exponenten zur Basis 10

S = Vorzeichen des Exponenten (S = 1 für negative Exponenten)

Fig. 6.3 : Codierung des WERT-Befehls

6.6.2.3 Der UPRO-Befehl

Dieser Befehl dient zur Markierung des Anfangs und zur Deklaration des Namens eines Unterprogramms. Unterprogramme müssen stets am Schluss des Hauptprogramms angefügt werden. Eine Unterprogrammdeklaration beginnt immer mit UPRO und hört mit ENDE auf. Der Name eines Unterprogramms besteht aus einem oder zwei Buchstaben, die intern mit je 5 Bits codiert werden.

Beispiel :

UPRO XY

Codierung:

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

										0	1	1	1	1	
								X	Y						

Fig. 6.4 : Codierung des UPRO-Befehls

6.6.2.4 Der ZIEL-Befehl

Der ZIEL-Befehl ist ähnlich wie der UPRO-Befehl aufgebaut und wird nur selten verwendet. Er dient zur Deklaration einer Marke als Sprungziel, deren Name auch im Binärcode noch erhalten bleibt. Im Binärcode ist in allen Sprungbefehlen (GUMP, WENN, RUFEN) das Sprungziel nur in Form einer Zeilennummer enthalten. Beim Rückübersetzen des Binärcodes kann das Übersetzungsprogramm jedoch bei allen Sprungbefehlen, welche auf ZIEL-Befehle führen, den ursprünglichen Namen des Sprungziels wieder rekonstruieren. Dies erhöht die Übersichtlichkeit des Programms, leider aber auch den Speicherplatzbedarf. ZIEL-Marken werden deshalb nur bei interaktiv entwickelten Lehrprogrammen benutzt. Bei auf Lochkarten geschriebenen Programmen ist der Einsatz gewöhnlicher Marken, die keinen zusätzlichen Speicherplatz benötigen (siehe Kap. 6.6.5), viel ökonomischer.

Der Name einer ZIEL-Marke besteht wie bei UPRO aus einem oder zwei Buchstaben, die intern mit je 5 Bits codiert werden.

Beispiel :

ZIEL YZ

Codierung :

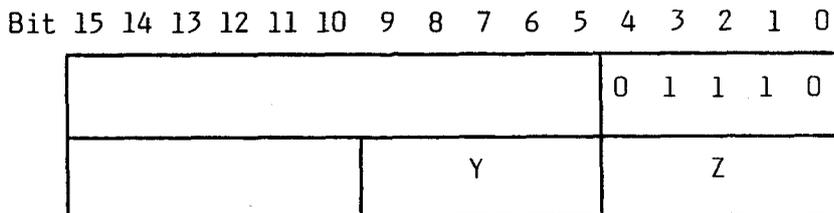


Fig. 6.5 : Codierung des ZIEL-Befehls

6.6.3 Die Befehle zur Ansteuerung der Konsolenperipherieeinheiten

6.6.3.1 Der BILD-Befehl

Der BILD-Befehl ist einer der häufigsten Befehle in einem Lehrprogramm. Er veranlasst die Projektoren A (links) und B (rechts) dazu, je ein Diapositiv anzuwählen und während einer bestimmten Zeit zu zeigen. Es gibt zwei Arten von BILD-Befehlen:

- Den normalen BILD-Befehl mit 3 Parametern für die oben beschriebenen Grundfunktionen.
- Den erweiterten BILD-Befehl mit 4 oder 5 Parametern, der neben den Grundfunktionen noch einige andere Aufgaben erfüllen kann.

Nach jedem BILD-Befehl wird der Programmablauf unterbrochen, bis eine Adressatenreaktion erfolgt oder die Zeit abläuft.

a) Der normale BILD-Befehl

Allgemeine Form : BILD n_L , n_R , z_v

n_L ist die Nummer des Dias, das auf dem linken Projektor gezeigt werden soll. Bereich: $0 \leq n_L \leq 80$.

n_R ist die Nummer des Dias, das auf dem rechten Projektor gezeigt werden soll. Bereich: $0 \leq n_R \leq 80$.

Das rechte Dia Nr. 1 ist als Titeldia für das Systemprogramm reserviert. Wenn $n_L = 0$ oder $n_R = 0$ ist, bleibt im betreffenden Projektor das alte Dia stehen.

z_v heisst Vorgabezeit und gibt an, wie lange die beiden Dias gezeigt werden sollen, wenn der Adressat keine Taste drückt. z_v besteht aus dem Buchstaben Z und einer Zahl, welche die gewünschte Anzahl Sekunden angibt. Die letzte Stelle dieser Zahl soll eine 0 sein, da die Zeit intern in 10-Sekunden-Einheiten dargestellt wird.

Wenn der Adressat die Taste ZUSATZZEIT drückt, erhält er zusätzlich nochmals die halbe Vorgabezeit zur Verfügung. Wie oft ZUSATZZEIT höchstens gedrückt werden kann, ist in der PROG-Anweisung (p_4) festgelegt. Die maximal mögliche Bearbeitungszeit für einen Lehrschritt beträgt 2500 Sekunden.

Beispiel :

BILD 16, 21, Z150

b) Der erweiterte BILD-Befehl

Allgemeine Form : BILD p_1 , p_2 , n_L , n_R , z_v

p_1 und p_2 sind die zusätzlichen Parameter. Einer von beiden kann fehlen.

$p_1 = A$: Die Antworttasten 1 - 5, WEITER und STOP sind wirksam, ohne dass der BILD-Befehl von einem Bewertungsbefehl gefolgt ist.

Codierung : Im Parameterwort ist Bit 15 = 1 .

$p_1 = B$: Die Konsole ist für die Dauer der Vorgabezeit blockiert. Alle Tasten sind unwirksam, Restzeitanzeige und akustisches Signal bei Zeitablauf werden unterdrückt. Der Adressat muss warten, bis die Zeit abgelaufen ist.

Codierung : Im Befehlswort ist Bit 7 = 1 .

$p_2 = S$: Das Tonbandgerät beginnt die gerade positionierte Tonsequenz abzuspielen.

Codierung : Im Parameterwort ist Bit 7 = 1 .

Beispiele :

BILD A, 59, 47, Z230

BILD B, S, 36, 69, Z80

BILD S, 14, 78, Z130

Codierung des ganzen Befehls :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

z_v in 10s-Einheiten				B		0	0	0	0	0
A	n_L			S	n_R					

Fig. 6.6 : Codierung des BILD-Befehls

6.6.3.2 Der BAND-Befehl

Dieser Befehl veranlasst das Aufsuchen und nötigenfalls das Abspielen einer bestimmten Tonsequenz durch das in die Lehrkonsole eingebaute Kassettentonbandgerät.

Man unterscheidet zwei Arten von BAND-Befehlen :

- Die initialisierenden BAND-Befehle mit zwei Parametern.
- Die den Programmablauf unterbrechenden BAND-Befehle mit 3 Parametern.

Eine bereits abgespielte Sequenz kann jeweils durch Drücken der Taste TONREPETITION erneut abgespielt werden.

a) Der initialisierende BAND-Befehl

Allgemeine Form : BAND p_1 , n_T

n_T ist die Nummer der gewünschten Tonsequenz ($0 \leq n_T \leq 127$). Sequenz Nr. 1 ist als Titelsequenz für das Systemprogramm reserviert. $n_T = 0$ bewirkt Rückspulen des Bandes an den Anfang.

$p_1 = S$: Das Tonbandgerät sucht den Anfang der Sequenz n_T und beginnt sie darauf sofort abzuspielen.

$p_1 = P$: Das Tonbandgerät soll das Band auf den Anfang der Sequenz n_T positionieren, diese jedoch nicht abspielen.

Beispiele:

BAND P, 17

BAND S, 123

Nach einem initialisierenden BAND-Befehl wird gleich der nächste Lehrprogrammbehl ausgeführt. Das Tonbandgerät sucht oder spielt dann selbständig die gewünschte Sequenz. Bei einem BAND S-Befehl hört der Adressat also die Tonsequenz dann, wenn die beiden Dias des auf BAND S folgenden BILD-Befehls gezeigt werden.

b) Der den Programmablauf unterbrechende BAND-Befehl

Allgemeine Form : BAND p_1 , n_T , z_v

n_T ist wieder die Nummer der gewünschten Tonsequenz ($0 \leq n_T \leq 127$)

z_v ist die Vorgabezeit analog wie beim BILD-Befehl und gibt an, wie lange bis zur automatischen Programmfortsetzung gewartet werden soll, wenn der Adressat keine Taste betätigt. Auch hier besteht z_v aus dem Buchstaben Z und der gewünschten Zeit in Sekunden.

Mögliche Werte des Parameters p_1 :

$p_1 = W$: Das Tonbandgerät sucht den Anfang der Sequenz n_T und spielt diese anschliessend ab. Darauf wird gewartet, bis der Adressat eine Taste drückt oder die Zeit abläuft. Aktiv und beleuchtet sind die Tasten WEITER und STOP.

$p_1 = A$: Wie W, aber zusätzlich sind die Antworttasten 1 - 5 aktiv und beleuchtet.

$p_1 = B$: Wie W, aber alle Tasten sind inaktiv, d.h. die Konsole ist gesperrt. In diesem blockierten Betrieb muss der Benutzer den Text des Bandes anhören und warten, bis die Zeit abläuft.

Beispiele :

BAND W, 43, Z50

BAND A, 56, Z30

BAND B, 92, Z120

Codierung des ganzen Befehls :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

z_v in 10s-Einheiten	C	T	L	0	1	0	0	1
	C	n_T						

Legende :

C : Steuerung des Abspielens. C = 1 für $p_1 = S, A, B$ und W

TL : Steuerung der Tastenlampen. TL = 00 für $p_1 = P$ und $p_1 = S$

TL = 01 für $p_1 = A$

TL = 10 für $p_1 = W$

TL = 11 für $p_1 = B$

n_T : Nummer der Tonsequenz ($0 \leq n_T \leq 127$)

z_v : Vorgabezeit

Fig. 6.7 : Codierung des BAND-Befehls

6.6.3.3 Der LAMP-Befehl

Dieser Befehl erlaubt das Ansteuern der fünf Meldungslampen FALSCH, FEHLMANIPULATION, ZUM ASSISTENTEN, RICHTIG und RESERVE .

Allgemeine Form : LAMP p_1, p_2, \dots (Maximal 5 Parameter)

Alle mit den Parametern p_i bezeichneten Lampen werden angezündet.

Als Parameter p_i sind zulässig :

F : FALSCH-Lampe anzünden, Bit 0 = 1 .

M : Lampe FEHLMANIPULATION anzünden, Bit 1 = 1 .

A : Lampe ZUM ASSISTENTEN anzünden, Bit 2 = 1 .

R : RICHTIG-Lampe anzünden, Bit 4 = 1 .

X : RESERVE-Lampe anzünden, Bit 5 = 1 .

0 : (Ziffer Null) Lampen von vorher ausgeführten LAMP-Befehlen löschen und nur die Lampen des aktuellen Befehls anzünden.

Codierung : Im Parameterwort ist Bit 15 = 1 .

Wenn mehrere LAMP-Anweisungen aufeinander folgen, werden die Parameter aller Befehle berücksichtigt (Ueberlagerung mit inklusiv-oder).

Die angezündeten Lampen verlöschen nach jedem Tastendruck oder Zeitablauf. Sollen sie weiterhin brennen, müssen sie neu angesteuert werden.

Die Bits 3, 6 und 7 sind für das Systemprogramm reserviert zur automatischen Steuerung der Tastenlampenbeleuchtung. Durch den Einsatz der RICHTIG-/FALSCH-Lampenautomatik (siehe Kap. 6.6.2.1) können in der Praxis die LAMP R- und LAMP F-Befehle meist eingespart werden.

Beispiele :

LAMP 0, X, A

LAMP M

Codierung :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

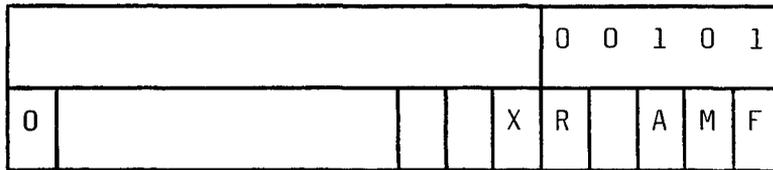


Fig. 6.8 : Codierung des LAMP-Befehls

6.6.3.4 Der ZEIG-Befehl

Dieser Befehl dient zur Darstellung einer reellen Zahl mit vierstelliger Mantisse und zweistelligem Exponenten auf der numerischen Anzeige der Konsole. Der Dezimalpunkt kann an beliebiger Stelle stehen. Der Exponent muss zwischen -63 und +63 liegen.

Beispiel :

ZEIG -35.68 E 47

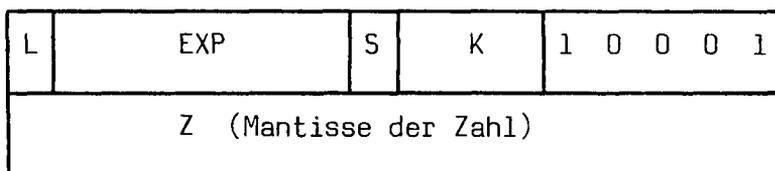
Mantisse Exponent zur Basis 10

Spezialfall:

ZEIG (ohne Parameter) löscht die numerische Anzeige, so dass nichts (BLANK) angezeigt wird.

Codierung :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



Legende :

- Z = Zahlenwert (Mantisse), ganzzahlig. Negative Zahlen werden als Zweierkomplement dargestellt. Bereich : -9999 ... +9999 .
- K = Anzahl Stellen hinter dem Dezimalpunkt (Komma). Bereich : 0 - 4.
- EXP = Betrag des Exponenten. Bereich : 0 - 63 .
- S = Vorzeichen des Exponenten. S = 1 für negative Exponenten.
- L = Anzeige-Lösch-Bit. L = 1, wenn ZEIG keinen Parameter hat.

Fig. 6.9 : Codierung des ZEIG-Befehls

6.6.4 Die Befehle für die Antwortbewertung

In den drei Befehlen BELA, BEVA und BENU werden eingegebene Antworten bewertet und auf dem Statistiklochstreifen registriert. Entsprechend der Vielfalt der möglichen Bewertungen können sie ziemlich viele Parameter besitzen.

Der erste Parameter nach dem Befehlswort ist bei allen drei Anweisungen der sogenannte Repetitionsfaktor r ($0 \leq r \leq 3$). Er gibt an, wie oft der Adressat eine Frage höchstens repetieren darf, wenn er beim ersten Versuch die richtige Antwort nicht findet. Eventuelle weitere Repetitionen werden von den Bewertungsbefehlen ignoriert und es wird für nachfolgende WENN-Befehle festgehalten, dass Repetitionsüberschreitung vorliegt. Falls der Repetitionsfaktor den Wert 0 hat, wenn also nur ein einziger Antwortversuch zulässig ist, darf er auch weggelassen werden.

Mit den beiden Befehlen BELA und BEVA werden Auswahlantworten bewertet. Die entsprechenden Tasten sind mit 1 - 5 bezeichnet. Jeder Auswahlantwort wird durch den Bewertungsbefehl eine bestimmte Antwortpunktzahl zugeordnet. Damit in den Bewertungsbefehlen keine unsinnigen

Beziehungen stehen (z.B. 2 = 3, wenn der Antwort 2 3 Punkte zugewiesen werden), werden die Auswahlantworten 1 - 5 in allen Lehrprogrammbefehlen wie folgt bezeichnet :

Antwort 1 : A

Antwort 2 : B

Antwort 3 : C

Antwort 4 : D

Antwort 5 : E

Auf den Fragedias werden die Auswahlantworten jedoch immer mit den entsprechenden Tastennummern markiert.

6.6.4.1 Der BELA-Befehl

Dieser Befehl dient zur Bewertung von Auswahlantworten in linearen Lehrprogrammteilen, bei denen keine Verzweigung auf Grund der gegebenen Antwort erfolgt. Bei richtigen Antworten, nach WEITER oder Zeitablauf wird anschließend der nächste Befehl ausgeführt. Bei falschen Antworten kann der Adressat dagegen weitere Antwortversuche machen, solange der Repetitionsfaktor nicht überschritten wird.

Allgemeine Form : BELA r, L, p₁ , p₂ , p₃ , p₄

r ist der Repetitionsfaktor (maximal zulässige Anzahl Wiederholungen, $0 \leq r \leq 3$). Maximale Anzahl Antwortversuche : $r + 1$.

L ist die Lösung und enthält die richtige(n) Antwort(en). Sind mehrere Antworten richtig, müssen sie durch Pluszeichen (+) getrennt werden.

p₁ Punktzahl für richtige Antwort im 1. Versuch

p₂ Punktzahl für richtige Antwort im 2. Versuch

p₃ Punktzahl für richtige Antwort im 3. Versuch

p₄ Punktzahl für richtige Antwort im 4. Versuch

Die Parameter p_i müssen ganzzahlig sein und die folgenden Bedingungen erfüllen :

$$0 \leq p_i \leq 15 \quad , \quad p_i > p_{i+1}$$

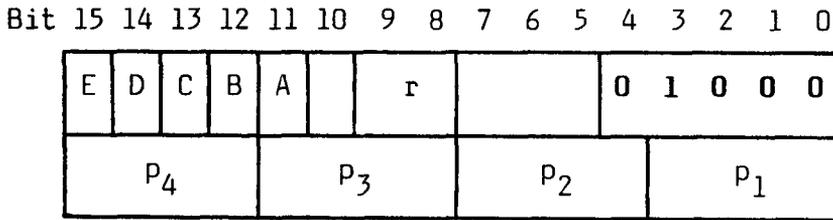
Parameter, die 0 sind, können weggelassen werden.

Beispiele :

BELA 2, A+D, 14, 10, 5

BELA 1, B, 15, 9

Codierung:



Die Bits E - A sind = 1, wenn die entsprechende Antwort richtig ist.

Fig. 6.10 : Codierung des BELA-Befehls

6.6.4.2 Der BEVA-Befehl

Dieser Befehl dient zur Bewertung von Auswahlantworten in verzweigten Lehrprogrammteilen, bei denen Verzweigungen in Funktion der gegebenen Antwort vorkommen. Nach der Ausführung einer BEVA-Anweisung wird immer der nächste Programmbefehl ausgeführt. Bei BEVA können alle Auswahlantworten mit verschiedenen Punktzahlen bewertet werden.

Allgemeine Form : BEVA r, P₁, P₂, P₃, P₄, P₅

r ist der Repetitionsfaktor (maximal zulässige Anzahl Wiederholungen, $0 \leq r \leq 3$). Maximale Anzahl Antwortversuche : r + 1 .

P_i (i = 1 - 5) sieht wie folgt aus :

$x = p_x$, wobei x = Betreffende Antwort (x = A, B, C, D oder E)
 p_x = Der Antwort x zugeordnete Punktzahl
 (ganzzahlig, $0 \leq p_x \leq 15$)

Parameter, die einer Antwort 0 Punkte zuweisen, können weggelassen werden.

Beispiele :

BEVA 2, A = 3 , C = 12 , D = 5

BEVA E = 15

Spezialfall :

BEVA (ohne Parameter)

Diese Anweisung bewertet die eingegebene Antwort nicht. Die Reaktion des Adressaten wird aber trotzdem auf dem Statistiklochstreifen registriert. Diese Sonderform der BEVA-Anweisung ist deshalb speziell für Umfragen geeignet.

Auch beim BEVA-Befehl besteht die Möglichkeit, mehrmals auf die gleiche Frage zu antworten. Nach einer falschen Antwort wird meist zunächst einige zusätzliche Information vermittelt. Anschliessend erfolgt mit den Anweisungen GUMP oder WENN ein Sprung auf den BILD-Befehl unmittelbar vor dem entsprechenden BEVA-Befehl. Die dem Adressaten gutgeschriebene Antwortpunktzahl wird jedoch bei w Wiederholungen mit dem Faktor $(0,75)^w$ multipliziert und ganzzahlig abgerundet.

Codierung :

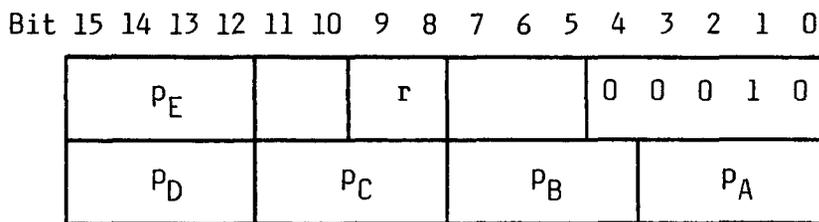


Fig. 6.11 : Codierung des BEVA-Befehls

6.6.4.3 Der BENU-Befehl für die Fehlerklasseneinteilung und Bewertung von numerischen Konstruktivantworten

Ein BENU-Befehl ist immer von zwei WERT-Anweisungen gefolgt, die das Intervall begrenzen, in dem eine vollständig richtige Antwort liegen muss. Es ist dabei gleichgültig, ob zuerst die obere oder die untere Schranke angegeben ist.

Jedem Resultat, das der Adressat eingibt, wird im BENU-Befehl zunächst eine bestimmte Fehlerklasse zugeordnet. Diese Fehlerklasse ist vom Resultat, von den Intervallgrenzen und von der Intervallbreite abhängig. Anschliessend erfolgt die Bewertung auf Grund der erzielten Fehlerklasse. Nach der Ausführung des BENU-Befehls fährt das Programm mit dem auf die beiden WERT-Anweisungen folgenden Lehrprogrammbefehl weiter. Mit entsprechenden WENN-Befehlen lassen sich darauf Programmverzweigungen auf Grund der erzielten Fehlerklasse realisieren.

Fig. 6.12 zeigt die Fehlerklasseneinteilung in Funktion des erzielten Resultats.

Lage des eingegebenen Resultats	Zugeordnete Fehlerklasse
$R < S-100A$	M4 (Fehlerklasse -4)
$S-100A \leq R < S-10A$	M3 (Fehlerklasse -3)
$S-10A \leq R < S-3A$	M2 (Fehlerklasse -2)
$S-3A \leq R < \text{MIN} = S-A$	M1 (Fehlerklasse -1)
$S-A = \text{MIN} \leq R \leq \text{MAX} = S+A$	F0 (Fehlerklasse 0)
$S+A = \text{MAX} < R \leq S+3A$	P1 (Fehlerklasse +1)
$S+3A < R \leq S+10A$	P2 (Fehlerklasse +2)
$S+10A < R \leq S+100A$	P3 (Fehlerklasse +3)
$S+100A < R$	P4 (Fehlerklasse +4)
$\text{MIN} \leq -R \leq \text{MAX}$	V (Vorzeichenfehler)
$\text{MIN} \leq 10^n R \leq \text{MAX} \quad *)$	K (Kommafehler)

*) Ein Kommafehler ist nur definiert, wenn 0 ausserhalb des durch MIN und MAX begrenzten Intervalls liegt.

Legende:

- R = Eingegebenes Resultat
- MIN = Untere Schranke für vollständig richtige Antwort
- MAX = Obere Schranke für vollständig richtige Antwort
- S = Sollwert = $\frac{1}{2}(\text{MAX} + \text{MIN})$
- A = Höchstzulässige Sollwertabweichung = $\frac{1}{2}(\text{MAX} - \text{MIN})$

Fig. 6.12 : Fehlerklasseneinteilung bei numerischer Bewertung

Resultate kleiner als der Sollwert erhalten also Minusfehlerklassen zugeteilt, Resultate grösser als der Sollwert dagegen Plusfehlerklassen. Komma- oder Vorzeichenfehler haben gegenüber normalen Fehlerklassen immer Priorität. Wenn beispielsweise ein Resultat mit Kommafehler nach obiger Definition auch noch in Fehlerklasse P3 liegt, so ist trotzdem nur die Fehlerklasse K gültig.

Im Bewertungsbefehl BENU können Plus- und Minusfehlerklassen gleicher Ordnung nur gemeinsam bewertet werden. Für die Bewertung werden die Fehlerklassen P1 und M1 zu einer Wertungsklasse F1, P2 und M2 zu F2, P3 und M3 zu F3 zusammengefasst. Daneben können auch Vorzeichen- und

Kommafehler bewertet werden. Im numerischen WENN-Befehl (siehe Kap.6.6.5.3) können die Minus- und Plusfehlerklassen jedoch unterschieden werden.

Allgemeine Form : BENU r, P₁, P₂, P₃, P₄, P₅, P₆

r ist der Repetitionsfaktor (Maximal zulässige Anzahl Wiederholungen, $0 \leq r \leq 3$). Maximale Anzahl Antwortversuche : r + 1 .

P_i (i = 1 - 6) sieht wie folgt aus :

x = p_x , wobei x = Betreffende Fehler- oder Wertungsklasse
(F0, F1, F2, F3, K oder V)

p_x = Der entsprechenden Klasse zugeordnete Punktzahl
(ganzzahlig, $0 \leq p_x \leq 15$) .

Parameter, in denen 0 Punkte zugewiesen werden, können weggelassen werden.

Es muss gelten : P_{F0} ≥ P_{F1} ≥ P_{F2} ≥ P_{F3} .

Beispiele :

BENU 1, F0 = 13, F1 = 4, K = 3, V = 1

BENU F0 = 11

Spezialfall :

BENU (ohne Parameter)

Diese Anweisung kann analog dem parameterlosen BEVA-Befehl für Umfragen verwendet werden. Eine Bewertung erfolgt nicht. Trotzdem wird das eingegebene Resultat auf dem Statistiklochstreifen festgehalten.

Codierung :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

P _V	P _K	r		0	1	0	1	1
P _{F3}	P _{F2}	P _{F1}		P _{F0}				

Fig. 6.13 : Codierung des BENU-Befehls

6.6.5 Die Befehle zur Steuerung des Programmablaufs

6.6.5.1 Der STOP-Befehl

Dieser Befehl dient zum Abbruch eines Lehrprogrammdurchlaufs. Es gibt zwei Arten von STOP-Befehlen: Den unbedingten und den bedingten STOP.

Der unbedingte STOP-Befehl hat keine Parameter und hat im Prinzip die gleiche Wirkung, wie wenn der Adressat die Taste STOP drückt. Der Hauptteil eines Lehrprogrammes (Hauptprogramm) wird praktisch immer durch einen solchen unbedingten STOP-Befehl abgeschlossen. Danach können noch Definitionen von Unterprogrammen folgen (mit UPRO-Befehl, siehe Kap. 6.6.2.3) .

Der bedingte STOP-Befehl hat folgende allgemeine Form :

STOP p_1 , p_2 , p_3

Er wird nur ausgeführt, wenn die durch p_1 bestimmte durchschnittliche Punktzahl pro Antwort ausserhalb des durch p_2 und p_3 begrenzten Intervalls (p_2 , p_3] liegt.

$$p_1 = \begin{cases} P, & \text{wenn die Antwortpunkte gemeint sind} \\ Z, & \text{wenn die Zeitpunkte gemeint sind} \\ T, & \text{wenn die Totalpunkte gemeint sind} \end{cases}$$

p_2 ist die untere Schranke für diese Durchschnittspunktzahl

p_3 ist die obere Schranke für diese Durchschnittspunktzahl

Die beiden Parameter p_2 und p_3 dürfen eine Kommastelle aufweisen. Sie können für $p_1 = P$ und Z im Bereich 0 - 15, für $p_1 = T$ dagegen im Bereich 0 - 25,5 liegen.

Regel für die Intervallgrenzen:

Die obere Grenze gehört } zum durch p_2 und p_3 bestimmten
Die untere Grenze gehört nicht } Intervall (p_2 , p_3] .

Ausnahme: Wenn $p_2 = 0$ ist, so gehört p_2 auch zum Intervall.

Bei einem STOP-Befehl wird der Grund (unbedingter STOP, Ueber- resp. Unterschreitung der entsprechenden Punktzahl beim bedingten STOP) auf dem Statistiklochstreifen festgehalten.

Beispiele :

STOP (unbedingter STOP, keine Parameter)

STOP T, 10.5, 14

STOP P, 8.5, 12.5

STOP Z, 0, 5.5

Codierung:

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

P ₁		0 0 0 0 1
P ₃		P ₂

Legende :

$$P_1 = \begin{cases} 00 \text{ f\u00fcr } p_1 = P \text{ (Antwortpunkte)} \\ 10 \text{ f\u00fcr } p_1 = Z \text{ (Zeitpunkte)} \\ 11 \text{ f\u00fcr } p_1 = T \text{ (Totalpunkte)} \end{cases}$$

P₂ = zehnfache untere Punktschranke p₂ (als ganze Zahl)

P₃ = zehnfache obere Punktschranke p₃ (als ganze Zahl)

Fig. 6.14 : Codierung des STOP-Befehls

6.6.5.2 Der GUMP-Befehl

Dieser Befehl bewirkt einen Sprung auf eine bestimmte Zeile.

Allgemeine Form : GUMP z

z ist das Sprungziel und kann entweder eine bestimmte Zeilennummer, eine relative Zeilennummer oder eine Marke sein.

Beispiele f\u00fcr absolute Zeilennummern :

GUMP 14

GUMP 381

Beispiele f\u00fcr relative Zeilennummern :

GUMP *+5

GUMP *-4

Der Stern bedeutet hier die aktuelle Zeilennummer der GUMP-Anweisung. Im oberen Beispiel wird also um 5 Zeilen vorw\u00e4rts, im untern um 4 Zeilen r\u00fcckw\u00e4rts gesprungen.

Beispiele f\u00fcr Marken als Sprungziele :

GUMP #XY

GUMP #C (# gibt an, dass eine Marke folgt)

Die Zeilen, auf die so gesprungen werden soll, m\u00fcssen mit einer entsprechenden Marke oder einem ZIEL-Befehl gekennzeichnet sein. Eine Marke besteht aus einem Namen mit einem oder zwei Buchstaben und einem Doppelpunkt.

Beispiele für mit Marken gekennzeichnete Programmanweisungen:

XY: BILD 47, 9, Z70

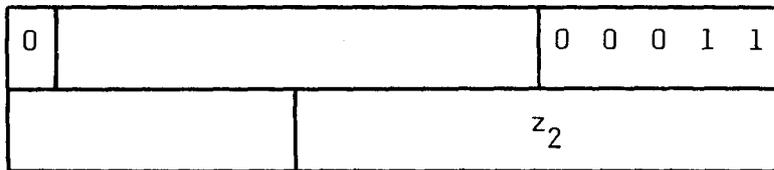
C: BAND W, 16

Die Verwendung von Marken als Sprungziele ist vor allem bei auf Lochkarten geschriebenen Lehrprogrammen günstig, da sie eine klare Gliederung des Programms ohne zusätzlichen Speicherplatzbedarf ermöglichen.

Beim fertig übersetzten Programm ist das Sprungziel immer als absolute Zeilennummer codiert. Bei der Verarbeitung von Marken als Sprungziele ersetzt das Uebersetzungsprogramm vor dem Ausstanzen des binären Lochstreifens alle Marken durch die entsprechende Zeilennummer. Damit Sprünge in den Programmkopf (erste 4 Worte des Lehrprogramms, durch PROG-Anweisung definiert) unmöglich sind, beginnt die interne Befehlsnumerierung des PLANETH-Systemprogramms erst beim ersten ausführbaren Befehl mit der Nummer 0. Da dieser Befehl jedoch bereits auf Zeile 2 steht, müssen alle Sprungziele bei der Codierung um 2 verkleinert werden.

Codierung :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



z_2 = Sprungziel als absolute Zeilennummer, um 2 verkleinert

Fig. 6.15 : Die Codierung des GUMP-Befehls

6.6.5.3 Der WENN-Befehl

Dieser Befehl bewirkt nur dann einen Sprung, wenn eine bestimmte Bedingung erfüllt ist. Diese Bedingung bezieht sich immer auf die zuletzt eingegebene Antwort. Falls die Bedingung nicht erfüllt ist, wird einfach der nächste Lehrprogrammbefehl ausgeführt.

Je nach der Art dieser Bedingung unterscheidet man drei Typen von WENN-Befehlen:

- a) WENN für Auswahlantworten und Antwortverzicht
- b) Fehlerklassen-WENN für numerische Antworten
- c) Intervall-WENN für numerische Antworten

Jeder WENN-Befehl hat folgende allgemeine Form : WENN G, z

z ist das Sprungziel und kann wie bei den GUMP-Befehlen eine absolute Zeilennummer, eine relative Zeilennummer oder eine Marke sein. Einzelheiten siehe Kap. 6.6.5.2 .

G umschreibt eine oder mehrere Bedingungen, von denen eine erfüllt sein muss, damit ein Sprung erfolgt. Wenn in G mehrere Bedingungen angegeben werden, so sind diese durch Pluszeichen (+ , entspricht logischem oder) zu trennen.

a) WENN für Auswahlantworten und Antwortverzicht

Die Ausführung eines Sprunges kann abhängig gemacht werden :

- (nur bei Fragen mit Auswahlantworten) von der Betätigung gewisser Antworttasten (A, B, C, D, E) .
- von der Betätigung der WEITER-Taste (W) .
- vom Ablaufen der Zeit (Z) .
- vom Ueberschreiten der zulässigen Repetitionszahl (R) .

Für die Sprungbedingung G sind die oben angegebenen Abkürzungen für die gewünschte(n) Bedingung(en) einzusetzen.

Beispiele :

WENN C, 139

WENN A+D, *-3

WENN W+Z+R, #AC

Codierung :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

W	Z	R			0	0			0	0	1	1	1		
E	D	C	B	A	z ₂										

W,Z,R,E,D,C,B,A = 1, wenn entsprechende Bedingung gewünscht, sonst 0

z₂ = Sprungziel als absolute Zeilennummer, um 2 verkleinert

Fig. 6.16 : Die Codierung von WENN-Befehlen für Auswahlantworten und Antwortverzicht

b) Numerisches Fehlerklassen-WENN

Mit diesem WENN können im Anschluss an numerische Bewertungen in BENU Sprünge in Funktion der Fehlerklasse der eingegebenen Zahl realisiert werden.

Für die Formulierung der Sprungbedingung G können alle in BENU berechneten Fehlerklassen verwendet werden : M4, M3, M2, M1, F0, P1, P2, P3, P4, K, V .

Die Definition dieser Fehlerklassen ist aus Fig. 6.12 in Kap. 6.6.4.3 ersichtlich.

Beispiele :

WENN FO, 169

WENN P1+P2+P3, *+7

WENN M1+M2+K+V, #QB

Codierung :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

V	K	P4	P3	P2	P1	0	1					0	0	1	1	1
FO	M1	M2	M3	M4		z ₂										

V,K,P4,P3,P2,P1 } = 1, wenn entsprechende Bedingung gewünscht, sonst 0
 FO,M1,M2,M3,M4 }

z₂ = Sprungziel als absolute Zeilennummer, um 2 verkleinert

Fig. 6.17 : Die Codierung des numerischen Fehlerklassen-WENN-Befehls

c) Numerisches Intervall-WENN

Bei diesem WENN erfolgt ein Sprung, wenn die zuletzt eingegebene Zahl in einem bestimmten Intervall liegt. Dieses wird durch die beiden WERT-Befehle begrenzt, die einem numerischen Intervall-WENN immer folgen. Für die Sprungbedingung G ist in diesem Fall immer I einzusetzen. Wie beim BENU-Befehl gehören die Grenzen noch zum Intervall.

Beispiele (mit den beiden zugehörigen WERT-Befehlen) :

WENN I, 238

WERT 5.5

WERT 6.5

WENN I, #ST

WERT -1.7E-11

WERT -2.3E-11

Codierung :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

							1	1					0	0	1	1	1
										z ₂							

z₂ = Sprungziel als absolute Zeilennummer, um 2 verkleinert

Fig. 6.18 : Die Codierung des numerischen Intervall-WENN-Befehls

6.6.5.4 Der IMIN-Befehl

Dieser Befehl ermöglicht Verzweigungen, die vom gesamten bisherigen Lernverhalten des Adressaten abhängen.

- Es sind Verzweigungen möglich in Funktion der
- durchschnittlichen Antwortpunktzahl pro Antwort.
 - durchschnittlichen Zeitpunktzahl pro Antwort.
 - durchschnittlichen Totalpunktzahl pro Antwort.

Der IMIN-Befehl enthält im Gegensatz zu den GUMP- und WENN-Befehlen kein Sprungziel, sondern verzweigt nur entweder auf die nächste oder übernächste Zeile. Ein IMIN-Befehl ist deshalb meist von einem GUMP-Befehl gefolgt.

Allgemeine Form : IMIN p_1 , p_2 , p_3 , p_4

$$p_1 = \begin{cases} P, & \text{wenn die Antwortpunkte gemeint sind} \\ Z, & \text{wenn die Zeitpunkte gemeint sind} \\ T, & \text{wenn die Totalpunkte gemeint sind} \end{cases}$$

p_2 ist die untere Schranke } für die gewünschte durchschnittliche
 p_3 ist die obere Schranke } Punktzahl pro Antwort

p_4 ist entweder *+1 oder *+2 und gibt an, wohin verzweigt werden soll, wenn die in p_1 spezifizierte Punktzahl pro Antwort im Intervall $(p_2, p_3]$ liegt (IMIN = Im Intervall). Falls diese Punktzahl nicht im Intervall liegt, wird auf die andere, in p_4 nicht angegebene Zeile verzweigt.

Die beiden Parameter p_2 und p_3 dürfen eine Kommastelle aufweisen. Sie können für $p_1 = P$ und Z im Bereich 0 - 15, für $p_1 = T$ im Bereich 0 - 25,5 liegen.

Regel für die Intervallgrenzen (analog STOP-Befehl) :

Die obere Grenze p_3 gehört } zum durch p_2 und p_3 bestimmten
 Die untere Grenze p_2 gehört nicht } Intervall $(p_2, p_3]$.

Ausnahme : Wenn $p_2 = 0$ ist, so gehört p_2 auch zum Intervall.

Beispiele :

IMIN P, 3.5, 7, *+1

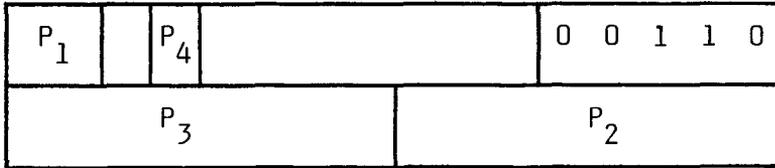
Wenn \bar{p} die durchschnittliche Antwortpunktzahl pro Antwort und $3.5 < \bar{p} \leq 7$ ist, wird auf die nächste Zeile, sonst auf die übernächste Zeile verzweigt.

IMIN Z, 0, 4.6, *+2

Wenn \bar{z} die durchschnittliche Zeitpunktzahl pro Antwort und $0 \leq \bar{z} \leq 4.6$ ist, wird auf die übernächste Zeile, sonst auf die nächste Zeile verzweigt.

Codierung :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



$$P_1 = \begin{cases} 00 & \text{für } p_1 = P \text{ (Antwortpunkte)} \\ 10 & \text{für } p_1 = Z \text{ (Zeitpunkte)} \\ 11 & \text{für } p_1 = T \text{ (Totalpunkte)} \end{cases}$$

$P_2 =$ zehnfache untere Punktschranke p_2 (als ganze Zahl)

$P_3 =$ zehnfache obere Punktschranke p_3 (als ganze Zahl)

$P_4 = 1$, wenn $p_4 = *+1$. 0, wenn $p_4 = *+2$.

Fig. 6.19 : Die Codierung des IMIN-Befehls

6.6.5.5 Der ZUFA-Befehl

Dieser Befehl dient zur Realisierung von zufälligen Verzweigungen. Er besitzt genau einen Parameter n ($2 \leq n \leq 15$), der die gewünschte Anzahl gleichwahrscheinlicher Zufallsverzweigungen angibt. Wie IMIN enthält auch ZUFA keine Sprungziele. Ein ZUFA-Befehl bewirkt auf Grund einer Zufallsgrösse einen Sprung auf einen der n folgenden Lehrprogramm-befehle.

Allgemeine Form : ZUFA n

Ein ZUFA n - Befehl ist praktisch immer von $n-1$ GUMP-Anweisungen gefolgt, damit die gewünschten Ziele angesprungen werden können.

Beispiel (inkl. zugehörige GUMP-Befehle) :

ZUFA 4
 GUMP #CB
 GUMP #OM
 GUMP 189

Codierung :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

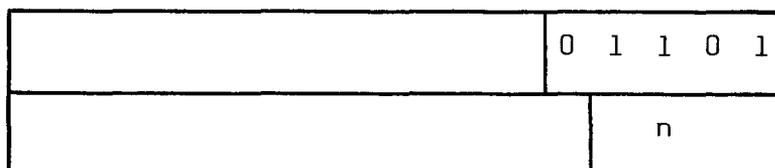


Fig. 6.20 : Die Codierung des ZUFA-Befehls

6.6.5.6 Der RUFEBefehl

Dieser Befehl dient zum Aufruf eines Unterprogramms vom Hauptprogramm aus. Eine Verschachtelung von Unterprogrammen ist nicht möglich, deshalb sind RUFEBefehle nur im Hauptprogramm erlaubt.

Allgemeine Form : RUFE u

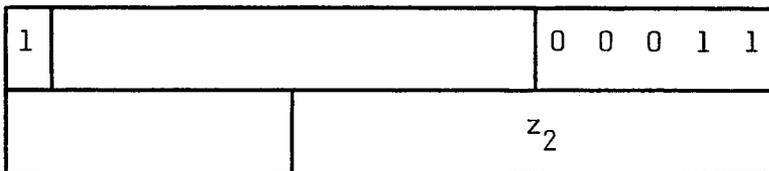
u = Name des gewünschten Unterprogramms (1 oder 2 Buchstaben, muss in einer UPRO-Anweisung deklariert sein).

Beispiel :

RUFE CW

Codierung :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



z_2 = Zeilennummer des zugehörigen UPRO-Befehls, um 2 verkleinert

Fig. 6.21 : Die Codierung des RUFEBefehls

6.6.5.7 Der ENDEBefehl

ENDE bewirkt den Rücksprung aus einem Unterprogramm ins aufrufende Hauptprogramm. Am Schluss jeder Unterprogrammdeklaration muss eine ENDE-Anweisung stehen. ENDE hat keine Parameter.

Beispiel :

(UPRO)

·
·

ENDE

Codierung :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

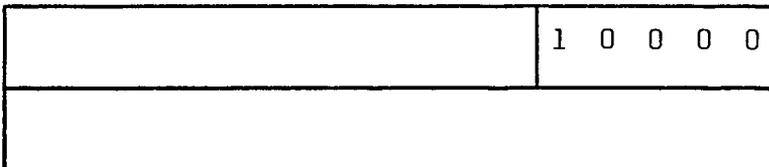


Fig. 6.22 : Die Codierung des ENDEBefehls

6.6.5.8 Der LEER-Befehl

Die LEER-Anweisung hat keinen Einfluss auf den Ablauf des Lehrprogramms, belegt aber gleichviel Speicherplatz wie alle andern Befehle. Die LEER-Anweisung wird gelegentlich benötigt, um Platz für zukünftige Programmweiterungen auszusparen. Bei der statistischen Auswertung der Antworten (siehe Kap. 7) erfolgt die Unterscheidung der einzelnen Fragen nämlich auf Grund der Zeilennummer. Bei Programmänderungen ist es deshalb günstig, wenn unveränderte Fragen weiterhin auf der gleichen Zeile stehen.

Beispiel :

LEER

Codierung :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

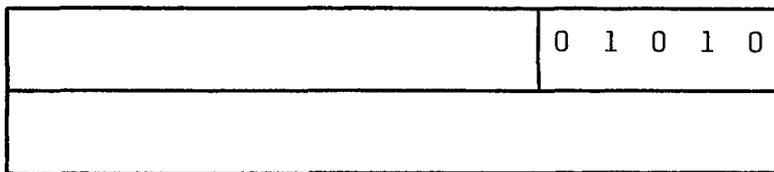


Fig. 6.23 : Die Codierung des LEER-Befehls

6.6.5.9 Der HALT-Befehl

Der HALT-Befehl bewirkt einen Unterbruch im Systemprogramm. Er darf nur zum Austesten verzweigter Lehrprogramme durch den Autor im l-Konsolenbetrieb verwendet werden. In fertig ausgetesteten Lehrprogrammen ist er streng verboten!

HALT hat einen Parameter p_H , der aus einer oder zwei Oktalstellen (fakultativ gefolgt vom Buchstaben B) besteht. Bei der Ausführung des Befehls wird der Computer angehalten und im T-Register erscheint der Code 1020XY, wobei XY die beiden Oktalstellen des Parameters p_H sind.

Allgemeine Form : HALT p_H

Beispiele :

HALT 17

HALT 26B

Codierung :

Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

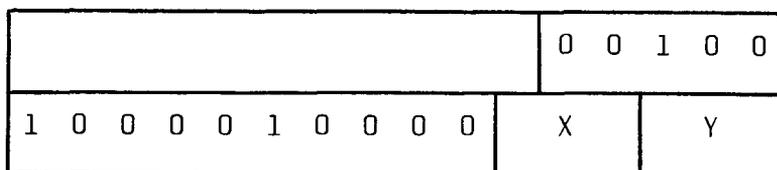


Fig. 6.24 : Die Codierung des HALT-Befehls

6.6.6 Lehrprogrammbeispiele

In diesem Kapitel werden drei Lehrprogrammbeispiele angegeben, die das Zusammenwirken der einzelnen Befehle und einige typische Programmstrukturen zeigen. Es handelt sich dabei um Demonstrationsprogramme, die auf wenigen Zeilen viele verschiedene Programmiermöglichkeiten darstellen sollen. In der Praxis verwendete Lehrprogramme besitzen wesentlich mehr Verzweigungen, sind beträchtlich länger und bestehen aus etwa 80 - 400 Zeilen. Maximal sind 1000 Zeilen zulässig.

Die folgenden drei Lehrprogramme wurden auf Lochkarten geschrieben. Um bei solchen Programmen eine übersichtliche Darstellung zu erhalten, können auch Kommentarzeilen verwendet werden, die mit einem Stern (*) in der ersten Kolonne beginnen.

6.6.6.1 Einfaches lineares Lehrprogramm ohne Ton

Für dieses unvertonte lineare Lehrprogramm nach Skinner [19] werden nur 4 verschiedene Befehle benötigt, nämlich PROG, BILD, BELA und STOP.

Im PROG-Befehl werden die RICHTIG-/FALSCH-Lampenautomatik und die Punkteanzeigeautomatik eingeschaltet, die maximale Anzahl Zusatzzeiten auf 2 und das Zeitgewicht auf 1 gesetzt.

```
0001      PROG 6661(F,P,2,1,0)      | R,F,P-AUTOM. EIN, AZZ=2, ZG=1
0002      BILD 2,2,Z100             | BEDIENUNGSANLEITUNG KONSOLE
0003      * BILD 10,3,Z150           | INFORMATION UND FRAGE 1
0004      BELA 1,B,13,5             | LIN. BEWERTUNG (1 REP)
0005      * BILD 12,4,Z180           | INFORMATION UND FRAGE 2
0006      BELA 1,C+E,12,7           | LIN. BEWERTUNG (1 REP)
0007      * BILD 13,5,Z120           | INFORMATION UND FRAGE 3
0008      BELA 1,D,10,4             | LIN. BEWERTUNG (1 REP)
0009      * BILD 14,8,Z220           | INFORMATION UND FRAGE 4
0010      BELA 1,A,15,6             | LIN. BEWERTUNG (1 REP)
0011      STOP                      | ENDE DES LEHRPROGRAMMS
/TTY
/NEW
```

Fig. 6.25 : Einfaches lineares Lehrprogramm ohne Ton

6.6.6.2 Einfaches verzweigtes Lehrprogramm ohne Ton

Dieses Programm besitzt nur Fragen mit Auswahlantworten. Es kommen darin Verzweigungen vor, die von der zuletzt gegebenen Antwort oder von der bisher erbrachten Gesamtleistung abhängen (Zeile 26).

Für die Programmierung genügen 8 verschiedene Anweisungen, nämlich PROG, BILD, BEVA, BELA, WENN, GUMP, IMIN und STOP .

```

0001      PROG 6662(F,P,3,2,0) ; R,F,P-AUTOM.EIN,AZZ=3,ZG=2
0002      BILD 0,2,2120      ; KONSOLENBEDIENUNGSANLEITUNG
0003      BILD 3,3,2180      ; EINFUEHRUNG IN PROGR.THEMA
0004      BILD 4,0,2200      ; THEORIE ZU FRAGE 1, 1.TEIL
0005      BILD A,0,4,2100    ; THEORIE ZU FRAGE 1, 2.TEIL
0006      WENN E,3           ; TASTE E:THEORIEWIEDERHOLUNG
0007      *
0007      * BILD 5,5,2150     ; FRAGE 1 ;
0008      BEVA 1,C=12        ; BEWERTUNG FRAGE 1 (1REP)
0009      WENN C+W+Z+R,12    ; RICHTIGE ANTW./VERZICHT: 12
0010      BILD 5,6,280      ; ERKLAERUNG ZU FALSCHANTW.
0011      GUMP 7             ; NEUER LOESUNGSVERSUCH
0012      BILD 7,7,2110     ; LOESUNG ZU FRAGE 1
0013      *
0013      BILD 8,8,2130     ; THEORIE ZU FRAGE 2
0014      * BILD 9,9,2120    ; FRAGE 2 ;
0015      BEVA 2,E=11,A=6,B=2 ; BEWERTUNG FRAGE 2 (2REP)
0016      WENN E+W+Z+R,25    ; RICHTIGE ANTW./VERZICHT: 25
0017      WENN A,21          ; WENN ANTW.A, SPRUNG AUF 21
0018      WENN B,23          ; WENN ANTW.B, SPRUNG AUF 23
0019      BILD 10,10,2200   ; ERKLAER. ZU FALSCHANTW. C+D
0020      GUMP 14           ; NEUER LOESUNGSVERSUCH
0021      BILD 0,11,260     ; ERKLAERUNG ZU ANTWORT A
0022      GUMP 14           ; NEUER LOESUNGSVERSUCH
0023      BILD 0,12,290     ; ERKLAERUNG ZU ANTWORT B
0024      GUMP 14           ; NEUER LOESUNGSVERSUCH
0025      BILD 12,13,2140   ; LOESUNG ZU FRAGE 2
0026      *
0026      IMIN P,0,6.5,*+1  ; LEISTUNGSABH. VERZWEIGUNG
0027      GUMP 33           ; SCHLECHTE ADRESSATEN:FRAGE4
0028      *
0028      *                   FUER GUTE ADRESSATEN ;
0028      BILD 14,14,2100   ; THEORIE ZU FRAGE 3
0029      * BILD 15,15,2200 ; FRAGE 3 : SCHWIERIGE FRAGE
0030      BELA 1,D,15,10    ; BEWERTUNG FRAGE3. D RICHTIG
0031      BILD 15,16,2240   ; LOESUNG ZU FRAGE 3
0032      GUMP 37           ; SPRUNG AUF PROGRAMMENDE
0033      *
0033      *                   FUER SCHLECHTE ADRESSATEN ;
0033      BILD 18,18,2150   ; THEORIE ZU FRAGE 4
0034      * BILD 19,19,2150 ; FRAGE 4 : LEICHTE FRAGE
0035      BELA 1,B,10,7     ; BEWERTUNG FRAGE4. B RICHTIG
0036      BILD 19,20,280    ; LOESUNG ZU FRAGE 4
0037      STOP              ; ENDE DES LEHRPROGRAMMS

```

Fig. 6.26 : Einfaches verzweigtes Lehrprogramm ohne Ton

6.6.6.3 Vertontes Lehrprogramm mit Auswahl- und Konstruktivantworten

```
0001      PRUS 6663(F,P,2,2,0)      ; R,F,P-AUTOM. EIN; A22=2; ZG=2
0002      BAND 5,2                    ;  BEDIENUNGSANLEITUNG
0003      BILD 2,2,Z100               ;  FUER LEHRMASCHINE
0004      *
0004      A:* BILD 3,3,Z120           ;  FRAGE 1
0005      BEVA E=13                  ;  BEWERTUNG; DANN VERZWEIGUNGEN
0006      WENN A:#AA                 ;  WENN ANTW.A; SPRUNG AUF AA
0007      WENN B:#AB                 ;  WENN ANTW.B; SPRUNG AUF AB
0008      WENN C:#AC                 ;  WENN ANTW.C; SPRUNG AUF AC
0009      WENN D:#AD                 ;  WENN ANTW.D; SPRUNG AUF AD
0010      WENN E:#AE                 ;  WENN ANTW.E; SPRUNG AUF AE
0011      GUMP #AW                   ;  SPRUNG AUF AW
0012      AA: BAND 5,3               ;  TONKOMMENTAR ZU ANTWORT A
0013      GUMP #AW                   ;  SPRUNG AUF AW
0014      AB: BAND 5,4               ;  TONKOMMENTAR ZU ANTWORT B
0015      GUMP #AW                   ;  SPRUNG AUF AW
0016      AC: BAND 5,5               ;  TONKOMMENTAR ZU ANTWORT C
0017      GUMP #AW                   ;  SPRUNG AUF AW
0018      AD: BAND 5,6               ;  TONKOMMENTAR ZU ANTWORT D
0019      GUMP #AW                   ;  SPRUNG AUF AW
0020      AE: BAND 5,7               ;  TONKOMMENTAR ZU ANTWORT E
0021      AW: BILD 4,4,Z150          ;  LOESUNG ZU FRAGE 1
0022      *
0022      BAND W,8,Z50              ;  EINFUEHRUNG ZU FRAGE 2+3
0023      ZUFA 2                     ;  2-FACHE ZUFALLSVERZWEIGUNG
0024      GUMP #C                    ;  SPRUNG AUF FRAGE 3
0025      *
0025      K:* BILD 5,5,Z80           ;  FRAGE 2
0026      BELA 2,A,12,8,4           ;  LIN. BEWERTUNG FRAGE 2 (2 REP)
0027      BILD 0,6,Z90              ;  LOESUNG ZU FRAGE 2
0028      GUMP #D                    ;  SPRUNG AUF FRAGE 4
0029      *
0029      C:* BILD 7,7,Z80           ;  FRAGE 3
0030      BELA 2,D,12,8,4           ;  LIN. BEWERTUNG FRAGE 3 (2 REP)
0031      BILD 0,8,Z90              ;  LOESUNG ZU FRAGE 3
0032      *
0032      D:* BILD 9,9,Z240          ;  FRAGE 4 (NUMERISCH)
0033      BENU F0=15,K=4,V=1        ;  NUMERISCHE BEWERTUNG
0034      WERT 149E3                 ;  UNTERE SCHRANKE
0035      WERT 151E3                 ;  OBERE SCHRANKE
0036      RUF E RK                    ;  RESULTAT KOMMENTIEREN
0037      BILD 10,10,Z160           ;  LOESUNG ZU FRAGE 4
0038      WENN F0:#U                 ;  WENN ANTW. RICHTIG; SPRUNG U
0039      *
```

Fig. 6.27a : Vertontes Programm mit Auswahl- und Konstruktivantworten

```

0039 E:* BILD 11,11,2270 ; FRAGE 5 (NUMERISCH)
0040 BENU 1,F0=14,F1=3,V=2; NUMERISCHE BEWERTUNG
0041 WERT -1.2E6 ; OBERE SCHRANKE
0042 WERT -1.3E6 ; UNTERE SCHRANKE
0043 WENN F0,#EB ; ANTW.RICHTIG,SPRUNG AUF EB
0044 WENN W+Z+R,#EB ; ANTWORTVERZICHT,SPRUNG EB
0045 WENN I,#EA ; SPRUNG AUF EA; WENN
0046 WERT -20E6 ; RESULTAT ZWISCHEN
0047 WERT -18E6 ; -20E6 UND -18E6 LIEGT
0048 RUFK RK ; RESULTAT KOMMENTIEREN
0049 BILD 0,13,2100 ; ERKLAERUNG ZU FALSCHANTW.
0050 GUMP #E ; NEUER LOESUNGSVERSUCH
0051 EA: BILD 13,0,2100 ; ERKLAERUNG ZU TYP. FEHLER
0052 GUMP #E ; NEUER LOESUNGSVERSUCH
0053 EB: BILD 12,12,2200 ; LOESUNG ZU FRAGE 5
0054 *
0054 U:* BILD 14,14,280 ; FRAGE6 (O.BEW): URTEIL DES
0055 BEVA ; ADRESSATEN UEBER LEHRPROG.
0056 *
0056 * ; SCHLUSSBEWERTUNG :
0056 IMJN T,10,17,*+1
0057 BILD 15,15,230 ; GUT GEARBEITET
0058 IMJN T,6.5,10,*+1
0059 BILD 15,16,230 ; MITTELMAESSIG GEARBEITET
0060 IMJN T,0,6.5,*+1
0061 BILD 15,17,230 ; SCHLECHT GEARBEITET
0062 *
0062 BILD B,18,18,230 ; SCHLUSSDIAS ZEIGEN
0063 STOP ; ENDE DES LEHRPROGRAMMS
0064 *
0064 UPRK RK ; UNTERPR. RESULTATKOMMENTAR
0065 WENN P1+M1,#RA ; KNAPP DANEBEN, AUF RA
0066 WENN K,#RB ; KOMMAFEHLER, AUF RB
0067 WENN V,#RC ; VORZEICHENFEHLER, AUF RC
0068 GUMP #RD ; RUECKSPRUNG
0069 RA: BAND S,9 ; TON: KNAPP DANEBEN
0070 GUMP #RD ; RUECKSPRUNG
0071 RB: BAND S,10 ; TON: KOMMAFEHLER
0072 GUMP #RD ; RUECKSPRUNG
0073 RC: BAND S,11 ; TON: VORZEICHENFEHLER
0074 RD: ENDE ; RUECKSPRUNG INS HAUPTPROG.

```

Fig. 6.27b : Vertontes Programm mit Auswahl- und Konstruktivantworten

Dieses Programm besitzt Fragen mit Auswahlantworten (Frage 1, 2, 3) und numerischen Konstruktivantworten (Frage 4 und 5). Neben nur antwortabhängigen Verzweigungen gibt es auch Verzweigungen auf Grund der Gesamtleistung. Sie werden hier zur Schlussbewertung des Adressaten eingesetzt (Zeile 56 - 61). Für die Steuerung des Tonbandgerätes werden meist BAND S - Befehle verwendet, bei denen der Ton beim darauf folgenden BILD-Befehl abgespielt wird (Ausnahme: BAND W auf Zeile 22). Das Unterprogramm RK dient zur akustischen Kommentierung von knapp neben dem richtigen Wert liegenden Resultaten, Vorzeichenfehlern und Kommafehlern bei Fragen mit numerischen Konstruktivantworten.

6.7 Die Herstellung der Diapositive

Die Vorlagen für die Herstellung der Bilder werden mit Schreibmaschine und farbigen Filzschreibern meist auf Karten im Format A6 oder A5 geschrieben. Damit projizierte Texte auf den Mattscheiben gut lesbar sind, sollte für Textvorlagen wenn möglich das Format A6 verwendet werden. Für Zeichnungen ist das Format A5 günstiger. Ganz komplizierte Zeichnungen können auch auf A4 ausgeführt werden.

Diese Vorlagen werden mit einer normalen Kleinbildkamera fotografiert. Ein speziell konstruiertes Stativ gewährleistet den richtigen Abstand für alle Vorlagen in den Formaten A4, A5 und A6. Die Beleuchtung erfolgt durch zwei seitlich angebrachte 150W-Fotolampen. Für farbige Bilder wird normaler Diafilm verwendet. Für schwarz-weiße Aufnahmen (Text) eignet sich ein besonders steiler Dokumentenfilm besser, der weiße und hellgraue Stellen weiss, schwarze und dunkelgraue Stellen dagegen schwarz abbildet. Dadurch sind die Anforderungen an die Qualität der Textvorlagen etwas weniger streng.

Wenn die Zeit drängt, können die Dias in Kartonrähmchen eingesetzt werden, also so, wie sie manche Filmentwicklungsanstalt liefert. Für sehr oft verwendete Programme ist es jedoch ratsam, die Dias in etwas solidere Plastikrähmchen mit Deckgläsern einzurahmen. Nebst einem besseren mechanischen Schutz gewinnt man dadurch auch eine bessere Bildqualität, da die Deckgläser Filmwölbungen verhindern.

Darauf müssen die eingerahmten Dias richtig in die 80-plätzigigen Rundmagazine eingeordnet werden. Auf jedem Bild ist deshalb in der linken oberen Ecke die Lehrprogrammnummer angegeben. In der rechten oberen Ecke steht ein Buchstabe (L für linken, R für rechten Projektor) und die Positionsnummer. Nach dem Einfüllen der Dias erfolgt dann die Schlusskontrolle durch den Programmautor.

Es ist oft günstig, in einem Magazinpaar zwei Lehrprogramme statt nur einem unterzubringen, da diese Programme dann gewisse Dias gemeinsam benützen können (Bedienungsanleitung, allgemeine Hinweise, Schlussbewertung, Meinungsumfrage über das Lehrprogramm usw.) . Ueberdies weisen Programme mit 35 - 45 Dias pro Magazin meist gerade die von der Mehrheit der Adressaten als ideal empfundene Dauer von 30 - 45 Minuten auf.

6.8 Die Tonaufnahmen

Bei der Herstellung von PLANETH-Tonkassetten werden auf der einen Spur, der Tonspur, die gewünschten Sprach- und Tonsequenzen aufgenommen. Auf der andern Spur, der Steuerspur, wird der Anfang der einzelnen Sequenzen mit einer Sequenzmarkierung nach Fig. 3.16 markiert (Einzelheiten siehe Kap. 3.11.1).

Diese Sequenzmarkierungen müssen immer die vorgeschriebene Form und einen minimalen Abstand von 15 Sekunden haben. Sie werden von einem Hilfsgerät erzeugt, das weitgehend gegen Fehlbedienung geschützt ist. Nach dem Drücken des Auslöseknopfes wird zunächst die HF- und nach einiger Zeit die NF-Marke erzeugt. Nach 9s, wenn die Sequenzmarkierung vorüber ist, leuchtet eine Meldungslampe "SPRECHEN" auf und es wird gleichzeitig das Mikrofon für die Aufnahme der Sequenz auf die Tonspur aktiviert. Erst nach weiteren 6s wird auch die Auslösetaste wieder freigegeben.

Da zu jedem Lehrprogramm meist mehrere Sätze von Lehrstoffspeichern hergestellt werden, ist das direkte Bespielen der Tonkassetten unrationell. Die Herstellung der Tonkassetten erfolgt deshalb in der Regel in zwei Phasen: Zuerst wird auf einem qualitativ hochwertigen Spulentonbandgerät mit 19cm/s Bandgeschwindigkeit ein Mutterband aufgenommen. Davon werden dann die benötigten Kopien auf Kassetten mit 4,75cm/s hergestellt. Auf diese Weise ist eine sichere Archivierung und eine gleichmässige Qualität der Tonaufnahmen gewährleistet.

7. DIE VOM PLANETH-SYSTEM GELIEFERTEN STATISTIKDATEN

UND IHRE AUSWERTUNG

Ein wesentlicher Vorteil von PLANETH gegenüber vergleichbaren Lehrsystemen ist die Möglichkeit einer eingehenden statistischen Analyse der Lehrprogrammdurchläufe. Sie liefert den Programmautoren eine ausgezeichnete Rückmeldung über die Qualität der Lehrprogramme und den Lernerfolg der Adressaten.

7.1 Das Konzept der PLANETH-Statistik

Während der Programmdurchläufe stanzt das PLANETH-Systemprogramm alle für eine spätere statistische Auswertung wichtigen Daten (Antwort, Zeilennummer der zugehörigen Frage, Zeitbedarf, erzielte Punktzahl) laufend auf Lochstreifen. Zu Beginn jedes Programmdurchlaufs wird die fakultativ eingetippte persönliche Nummer des Adressaten (ETH-Nummer), die Lehrprogrammnummer und das Datum ausgegeben; am Schluss folgen Abschlusszeit, Bearbeitungsdauer und einige Hilfsdaten.

Die anfallenden Statistikdaten werden jeweils in Gruppen von 5 Bytes (40 Bits) ausgestanzt. Eine solche zusammenhängende 5-Byte-Gruppe wird als "Statistikstanzung" bezeichnet. Für die Ausgabe der Anfangsdaten (ETH-Nummer, Lehrprogrammnummer und Datum) werden zusammen zwei solche Statistikstanzungen benötigt. Auch für die Registrierung einer numerischen Antwort sind zwei Statistikstanzungen erforderlich. Für eine Auswahlantwort und für die Ausgabe der Schlussdaten genügt dagegen eine Statistikstanzung.

Die bei einem Lehrprogrammdurchlauf auftretende Anzahl Statistikstanzungen beträgt somit

$$a_{ss} = a_A + 2a_N + 3 \quad (7.1)$$

wobei a_{ss} = Anzahl Statistikstanzungen dieses Lehrprogrammdurchlaufs

a_A = Anzahl Auswahlantworten dieses Lehrprogrammdurchlaufs

a_N = Anzahl numerische Antworten dieses Lehrprogrammdurchlaufs

Dies entspricht $5a_{ss}$ Bytes oder $40a_{ss}$ Bits. In der Praxis ergibt ein Programmdurchlauf eine Datenmenge von etwa 50 - 400 Bytes (400 - 3200 Bits).

Es ist sicher nicht zweckmässig, derart grosse Datenmengen im Kernspeicher des Computers (16K Bytes, vergl. Kap. 4.1) zu behalten. Die einzelnen Statistikstanzungen werden deshalb fortlaufend auf Lochstreifen ausgegeben.

Da die einzelnen Antworten bei gleichzeitigem Betrieb mehrerer Lehrkonsolen stochastisch erfolgen, sind diese Statistikstanzungen völlig ungeordnet, d.h. zwischen die zu einer bestimmten Konsole gehörenden Statistikstanzungen sind solche anderer Konsolen eingestreut.

Gemäss dem Konzept des Lehrsystems PLANETH soll die Auswertung der Statistikdaten mit dem gleichen Computer vorgenommen werden können, der normalerweise den Lehrbetrieb steuert. Damit dies trotz der beschränkten Speicherkapazität des verwendeten Computers möglich ist, werden die Statistikdaten mit dem Statistikconverterprogramm STACO zuerst so geordnet, dass die Statistikstanzungen jedes Programmdurchlaufs sich unmittelbar folgen und zusammen einen Datensatz bilden.

Bei dieser Neuordnung entstehen aus den vielen relativ kurzen Lochstreifen der einzelnen Übungstage wenige lange, geordnete Statistiklochstreifen, welche die Daten aller in einem bestimmten Zeitraum (z.B. einem Monat) absolvierten Programmdurchläufe enthalten. Diese langen Lochstreifen lassen sich viel leichter archivieren und bei der anschliessenden Datenauswertung viel rationeller handhaben als die grosse Zahl kurzer Lochstreifen.

Das PLANETH-Statistikauswerteprogramm PLAST verarbeitet die von STACO geordneten Statistiklochstreifen und druckt die Resultate der Auswertung wahlweise auf dem Fernschreiber oder (wenn vorhanden) auf dem schnelleren Liniendrucker aus.

Es bestehen folgende Auswertungsmöglichkeiten :

a) Kurze Adressatenstatistik

Es werden die Adressatennummern (ETH-Nummern) und die Hauptdaten jedes Programmdurchlaufs ausgedruckt. Diese entsprechen etwa den jeweils im Schlussprotokoll ausgegebenen Daten.

Sonderfälle : Adressatenstatistik nur für ein bestimmtes Lehrprogramm mit automatischer Notengebung oder nur für bestimmte Adressatennummern.

b) Ausführliche Adressatenstatistik

Zusätzlich zur kurzen Adressatenstatistik wird jede Antwort mit zugehöriger Punktzahl, Antwortzeit und Zeilennummer der entsprechenden Frage ausgedruckt. Bei numerischen Antworten wird überdies das eingetippte Resultat ausgegeben.

c) Kurze Lehrprogrammstatistik :

Für ein bestimmtes Lehrprogramm werden die wichtigsten Gesamtdaten, wie Anzahl Lehrprogrammdurchläufe, durchschnittliche Totalpunktzahl, durchschnittliche Totalpunktzahl pro Antwort, Verteilung der durchschnittlichen Totalpunktzahl pro Antwort, mittlere Gesamtdauer der Programmdurchläufe, mittlere verbrauchte Zeit pro Antwort, Verteilung der Stopkriterien usw. ausgedruckt. Neben den Mittelwerten werden oft auch noch Maximal- und Minimalwerte angegeben.

d) Ausführliche Lehrprogrammstatistik :

Ausser den Angaben der kurzen Lehrprogrammstatistik wird zu jeder einzelnen Frage die Verteilung der Antworten, die mittlere erzielte Punktzahl und der durchschnittliche Zeitverbrauch ausgedruckt. Diese Angaben sind besonders wertvoll für die Beurteilung der Qualität jedes Fragelehrschritts.

Alle diese Auswertungen können eingeschränkt werden auf eine Auslese von Programmdurchläufen, die bestimmte Bedingungen erfüllen, beispielsweise auf jene, die in ein bestimmtes Zeitintervall (z.B. Sommersemester 1975) fallen, deren mittlere Punktzahl pro Antwort innerhalb gewisser Schranken liegt und die eine gewisse Minimaldauer und mindestens eine bestimmte Anzahl Antworten aufweisen. Weitere Einzelheiten über Aufbau und Möglichkeiten von PLAST befinden sich in Kap. 7.4 .

Dank diesen statistischen Auswertungen kann der Lehrprogrammautor genau sehen, wie sein Programm bei den Adressaten ankommt, ob es als ganzes oder nur in einzelnen Lehrschritten zu leicht oder zu schwierig ist und ob die vorgesehene Bearbeitungszeit richtig ist. Der Dozent, der Lehrprogramme in seinem Unterricht einsetzt, erhält einen gründlichen Einblick in das Lernverhalten jedes einzelnen Adressaten. Eine solche Fülle von Rückmeldungen von den Lernenden zu den Lehrenden kann mit konventionellen Unterrichtsmethoden niemals erreicht werden.

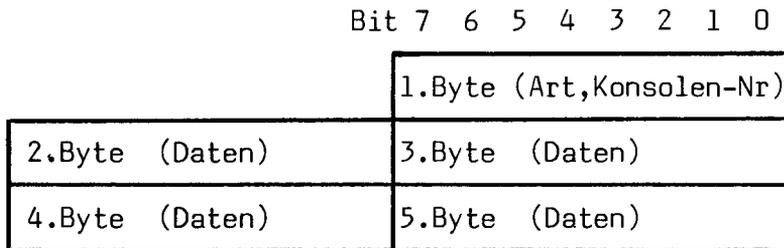
7.2 Die vom PLANETH-Systemprogramm gelieferten Daten

In diesem Kapitel werden die vom PLANETH-Systemprogramm ausgegebenen Statistikdaten genau beschrieben. Entsprechend den verschiedenen Fällen, die in der Statistik registriert werden müssen, gibt es 7 verschiedene Typen von Statistikstanzungen.

Die von den einzelnen Lehrprogrammdurchläufen stammenden Statistikdaten müssen beim Sortieren irgendwie unterschieden werden können. Dies geschieht mit Hilfe der Konsolenummer, die im ersten Byte jeder Statistikstanzung enthalten ist.

Im gleichen Byte ist in den Bits 7 - 5 auch die Art der Statistikstanzung angegeben.

Fig. 7.1 zeigt das prinzipielle Format einer Statistikstanzung. Die 5 Bytes werden unmittelbar hintereinander der Reihe nach ausgestanzt.



Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Fig. 7.1 : Prinzipielles Format einer Statistikstanzung

Zum Schutz der Statistikdaten gegen Uebertragungsfehler werden sie zu Datensätzen zusammengefasst, die mit einer Prüfsumme gesichert sind. Ein solcher Datensatz darf maximal 100 Statistikstanzungen umfassen, damit er von STACO (siehe Kap. 7.3) noch verarbeitet werden kann. Nach jedem Abschluss eines Lehrprogrammdurchlaufs wird der gerade aktuelle Datensatz abgeschlossen und mit einer speziellen Statistikstanzung die entsprechende Prüfsumme ausgegeben. Dies geschieht auch dann, wenn ein Datensatz auf die maximal zulässige Länge von 100 Statistikstanzungen angewachsen ist.

Infolge des beschränkten Computerspeicherplatzes darf ein Lehrprogrammdurchlauf höchstens 88 Statistikstanzungen umfassen, damit diese von STACO noch einwandfrei geordnet werden können. Das PLANETH-Systemprogramm überwacht deshalb laufend die Anzahl der Statistikstanzungen jedes Programmdurchlaufs. Wenn schon 87 Statistikstanzungen ausgegeben wurden, wird er automatisch mit einer speziell codierten Schlusstanzung abgebrochen. In der Praxis kommen derart lange Durchläufe nicht vor, so dass diese Sicherheitsmassnahme keine spürbare Einschränkung zur Folge hat.

7.2.1 Die Anfangsstanzen

Zu Beginn jedes Programmdurchlaufs werden mit zwei unmittelbar aufeinander folgenden Statistikstanzen folgende Grössen registriert :

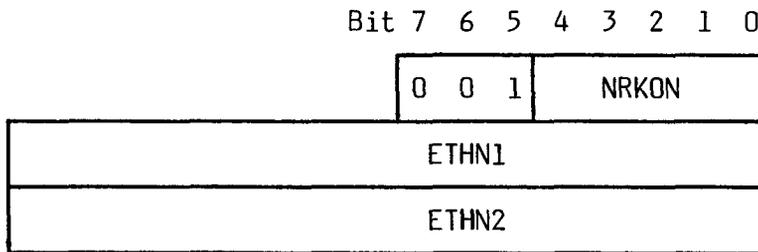
NRKON = Konsolenummer, Bereich 0 - 31

ETHN1 = Erste (linke) vier Stellen der Adressatenummer

ETHN2 = Letzte (rechte) vier Stellen der Adressatenummer

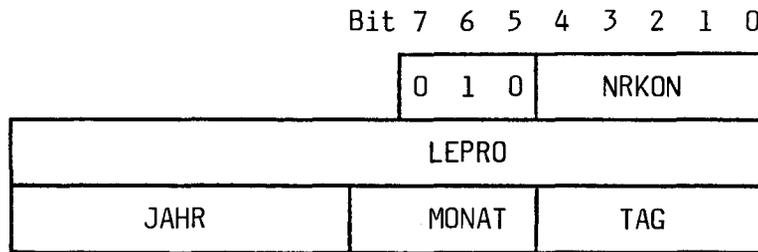
LEPRO = Lehrprogrammnummer (1 - 9999)

JAHR = }
MONAT = } Datum des Programmdurchlaufs (Jahr: 0 - 99)
TAG = }



Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Fig. 7.2 : Format der Anfangsstation Typ 1



Bit 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Fig. 7.3 : Format der Anfangsstation Typ 2

7.2.2 Die Auswahlantwortstation

Nach jeder Auswahlantwort werden folgende Grössen ausgestanzt :

NRKON = Konsolenummer, Bereich 0 - 31

ANTW = Antwort. 1 - 5 = Tasten 1 - 5, 14 = WEITER-Taste, 15 = Zeitablauf

PRZ = Programm-Zeilenummer der zugehörigen Frage (0 - 1000)

AP = Erzielte Antwortpunkte (0 - 15)

ZP = Erzielte Zeitpunkte (0 - 15)

ZEIT = Verbrauchte Antwortzeit in 10-Sekunden-Einheiten (0 - 250)

OB = 1, wenn keine Bewertung vorliegt, sonst 0

RE = 1, wenn diese Antwort eine Repetition ist, sonst 0

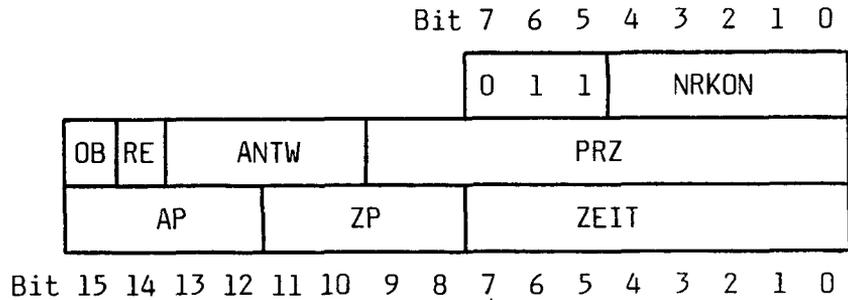


Fig. 7.4 : Format der Auswahlantwortstanzung

7.2.3 Die Statistikstanzungen bei numerischen Antworten

Nach jeder numerischen Antwort wird mit einer ersten Stanzung das Ergebnis der numerischen Bewertung registriert. Diese Fehlerklassenstanzung ist praktisch gleich aufgebaut wie eine Auswahlantwortstanzung. Darauf wird mit einer zweiten Stanzung noch das eingegebene Resultat festgehalten.

Neben den bei der Auswahlantwortstanzung beschriebenen Grössen werden noch folgende Grössen registriert :

FEKLA = Fehlerklasse (statt ANTW), in Fig. 6.12 definiert.

- Codierung :
- 0 = M4
 - 1 = M3
 - 2 = M2
 - 3 = M1
 - 4 = F0
 - 5 = P1
 - 6 = P2
 - 7 = P3
 - 8 = P4
 - 9 = K
 - 10 = V
 - 14 = WEITER-Taste
 - 15 = Zeitablauf

Z = Eingegebene reelle Zahl (Resultat). Die Registrierung erfolgt im normalen Format für reelle Zahlen (2 Worte zu 16 Bits), wobei:

- MANT = Mantisse des eingegebenen Resultates Z
- MSB = Stellen mit dem grösseren Gewicht
- LSB = Stellen mit dem kleineren Gewicht
- EXP = Exponent des eingegebenen Resultates Z
- V_M, V_E = Vorzeichen der Mantisse resp. des Exponenten

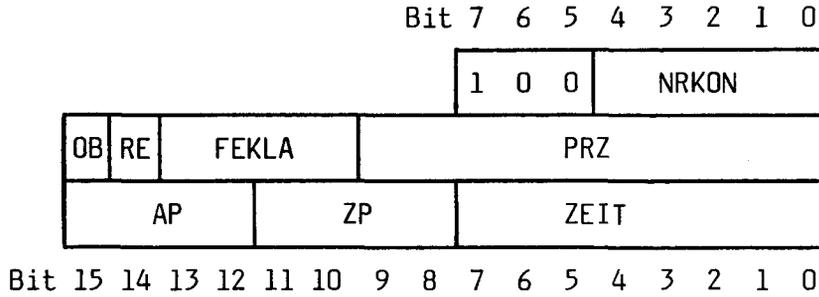


Fig. 7.5 : Format der Fehlerklassenstanzung

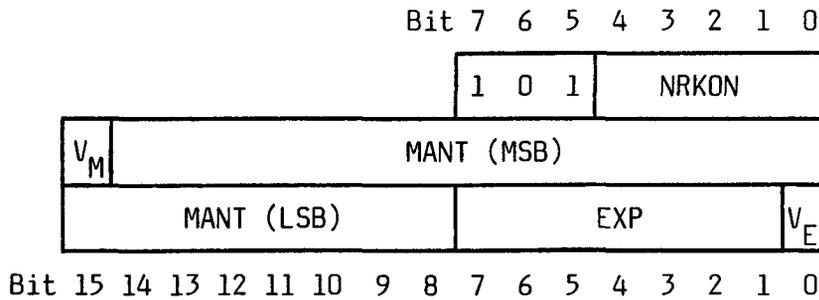


Fig. 7.6 : Format der Resultatstanzung

7.2.4 Die Schlusstanzung

Am Schluss jedes Lehrprogrammdurchlaufs werden folgende Größen ausgestanzt :

NRKON = Konsolenummer , Bereich 0 - 31

SK = Stopkriterium = Grund für Abbruch des Programmdurchlaufs

SK = 0 : Programmende (unbedingter STOP)

SK = 1 : STOP-Taste gedrückt

SK = 2 : Untere Punktlimite unterschritten bei bedingtem STOP-Befehl

SK = 3 : Obere Punktlimite überschritten bei bedingtem STOP-Befehl

SK = 4 : Systemstop vor Ausschalten des Systemprogramms

SK = 5 : Notstop wegen zu vieler Antworten (87 Stanzungen erreicht)

ZG = Zeitgewicht für die Berechnung der Totalpunkte, siehe Kap. 6.6.2.1
(ZG = Parameter p₅)

DAUER = Dauer des Lehrprogrammdurchlaufs in Minuten (0 - 1023)

STD }
MIN } = Uhrzeit bei Abschluss des Programmdurchlaufs

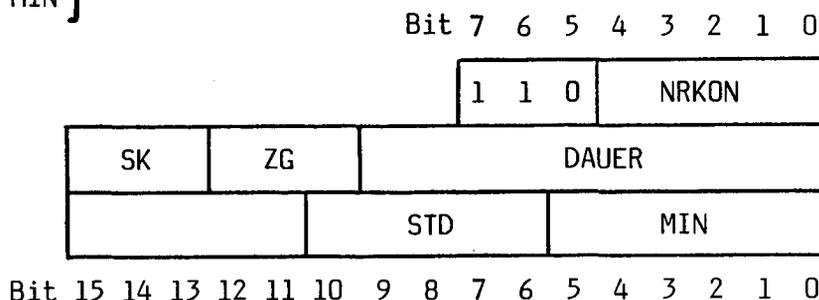


Fig. 7.7 : Format der Schlusstanzung

7.2.5 Die Prüfsummenstanzung am Schluss eines Datensatzes

Während der Ausgabe der Statistikstanzungen wird zu Kontrollzwecken laufend die Summe aller ausgestanzten Bytes des aktuellen Datensatzes berechnet. Diese Summe wird am Schluss des Datensatzes mit der Prüfsummenstanzung auf dem Lochstreifen festgehalten. Anschliessend werden 5 leere Bytes aus lauter Nullen ausgestanzt, damit die einzelnen Datensätze auf dem Lochstreifen auch optisch gut getrennt sind. Es ergibt sich somit ein totaler Abstand von 7 leeren Bytes zwischen den einzelnen Datensätzen.

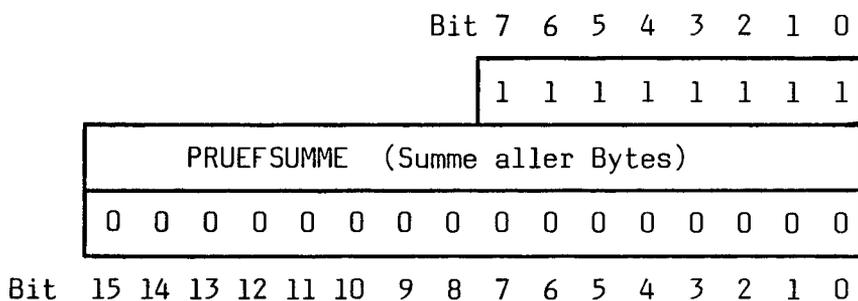


Fig. 7.8 : Format der Prüfsummenstanzung

7.3 Das Statistikconverterprogramm STACO

STACO hat die Aufgabe, aus den ungeordnet anfallenden Statistikstanzungen des PLANETH-Systemprogramms die Statistikdaten der einzelnen Lehrprogrammdurchläufe herauszusortieren.

Wenn alle Statistikstanzungen eines Programmdurchlaufs eingelesen worden sind, werden sie jeweils gesamthaft als ein Datensatz auf den geordneten (convertierten) Lochstreifen ausgestanzt.

Im Maximum können Statistikstanzungen von 32 Konsolen durcheinander gemischt sein. Damit STACO die Daten der einzelnen Konsolen sicher voneinander trennen kann, sind also 32 Pufferbereiche für Lehrprogramm-Statistikdaten erforderlich.

In STACO werden mit relativ langsamen Ein- und Ausgabemedien (siehe Kap. 4.2) grosse Datenmengen umgesetzt. Der eigentliche Sortiervorgang nimmt dabei nur sehr wenig Zeit in Anspruch. Es ist deshalb zweckmässig, das Einlesen der Daten mit dem Lochstreifenleser und die Ausgabe der geordneten Daten auf dem Lochstreifenstanzer im Interruptbetrieb vorzunehmen.

Der Sortiervorgang verläuft im Prinzip wie folgt :

Sobald eine Anfangsstanzung einer bestimmten Konsole eingelesen worden ist, wird dieser Konsole ein freier Pufferbereich für die Datenaufbereitung zugewiesen. Dort werden die Daten dieser Konsole nach ihrem Eintreffen jeweils abgespeichert. Sobald die Schlussstanzung der betreffenden Konsole registriert worden ist, wird die Ausgabe der in diesem Pufferbereich befindlichen Daten veranlasst, indem der Pufferbereich in die Ausstanz-Warteschlange eingereiht wird. Diese Warteschlange wird im Interruptbetrieb nach und nach abgearbeitet. Erst wenn alle Daten eines Pufferbereichs ausgestanzt sind, wird der Pufferbereich wieder für weitere Sortiervorgänge freigegeben.

Die einzelnen Pufferbereiche bieten Platz für 88 Statistikstanzungen und sind den einzelnen Konsolen nicht fest zugeteilt, damit sie optimal ausgenützt werden können. Zu einem bestimmten Zeitpunkt ist es ohne weiteres möglich, dass manche Konsolen keinen Pufferbereich benützen, während andere Konsolen zwei Bereiche belegen, nämlich einen zum Sortieren der von der Konsole gerade ankommenden Daten und einen zweiten für die fertig geordneten Daten eines auch von dieser Konsole stammenden Programmdurchlaufs, welche noch nicht oder nicht vollständig ausgestanzt worden sind.

Da der Lochstreifenleser schneller arbeitet als der Lochstreifenstanzer, kann es vorkommen, dass nach einiger Zeit keine freien Pufferbereiche mehr vorhanden sind. In diesem Fall wird das Einlesen unterbrochen und gewartet, bis wieder genügend freier Platz vorhanden ist.

Die einzelnen Statistikstanzungen werden von STACO nicht verändert. Ganz kurze Lehrprogrammdurchläufe mit weniger als 6 Statistikstanzungen werden jedoch von STACO unterdrückt. Dies geschieht auch mit unvollständigen Programmdurchläufen, die nicht ordnungsgemäss mit zwei Anfangsstanzen beginnen und mit einer Schlussstanzung aufhören. Auf diese Weise werden irrelevante und unvollständige Daten eliminiert, welche die spätere Auswertung stören könnten.

Auf dem von STACO ausgegebenen Lochstreifen sind die Datensätze, welche jeweils die Statistikstanzungen eines ganzen Lehrprogrammdurchlaufs umfassen, chronologisch nach der Uhrzeit beim Programmabschluss geordnet. Zur eindeutigen Unterscheidung von geordneten und ungeordneten Statistiklochstreifen wird das erste Byte der Prüfsummenstanzen, die einen solchen Datensatz abschliesst, bei geordneten Streifen durch 8 Nullen markiert statt durch 8 Einer wie bei ungeordneten Streifen (siehe Fig. 7.8). Die Prüfsumme selbst besteht in beiden Fällen aus der Summe aller Bytes des Datensatzes. Nach der Prüfsummenstanzen folgen wieder 5 leere Bytes aus lauter Nullen, damit die einzelnen Datensätze auch auf dem geordneten Statistikstreifen durch 7 leere Bytes getrennt sind.

7.4 Das PLANETH-Statistikauswerteprogramm PLAST

Dieses Programm wurde zur Reduktion des Programmieraufwandes in ALGOL geschrieben, einer höheren Programmiersprache für technisch-wissenschaftliche Anwendungen. Es verarbeitet nur von STACO geordnete PLANETH-Statistiklochstreifen und druckt die Resultate der Auswertung auf Fernschreiber oder Liniendrucker aus. Infolge der beschränkten Grösse des verfügbaren Computerspeichers (siehe Kap. 4.1) konnte bei der Resultatausgabe kommentierender Text nur sparsam verwendet werden. Dank einer übersichtlichen Darstellung sind die Ergebnisse der Auswertung jedoch trotzdem leicht verständlich.

Das Funktionsprinzip von PLAST ist einfach :

Nacheinander werden die einzelnen Datensätze eingelesen, die alle Statistikdaten eines bestimmten Lehrprogrammdurchlaufs enthalten. Zuerst werden daraus die für die Auswertung wichtigen Sortierdaten berechnet: Adressatennummer (ETH-Nr.), Lehrprogrammnummer, Datum, Dauer, Anzahl bewertete Antworten und mittlere Totalpunktzahl pro Antwort. Nur wenn alle diese Sortierdaten innerhalb gewisser Schranken liegen, wird der Programmdurchlauf ausgewertet. Die Schranken (Auswertekriterien) für dieses "Programmfilter" werden zu Beginn jeder Auswertung mit dem Fernschreiber eingegeben.

Fig. 7.9 zeigt das Grobflussdiagramm von PLAST. Nach dem Start des Programms wird durch die Eingabe der Steueranweisung die Art der gewünschten Statistikauswertung (Adressatenstatistik oder Lehrprogrammstatistik) definiert. In der Steueranweisung wird auch festgelegt, für welche der oben angegebenen Sortierdaten anschliessend Schranken eingelesen werden sollen, ob nur Erstantworten oder auch Repetitionen berücksichtigt werden sollen, ob eine kurze oder eine ausführliche Statistik gewünscht wird, ob bei einer Adressatenstatistik eine automatische Notengebung erfolgen soll und mit welchem Ausgabemedium die Resultate ausgedruckt werden sollen.

Nach dieser Steueranweisung müssen mit dem Fernschreiber die erforderlichen Auswertekriterien eingegeben werden. Beispielsweise ist anzugeben, in welchem Jahrgangs-Intervall die Adressatennummer liegen muss, ob auch anonym absolvierte Programmdurchläufe berücksichtigt werden sollen, in welchem Zeitintervall das Datum und in welchen Schranken die mittlere Totalpunktzahl liegen muss, welche Minimaldauer und welche minimale Antwortzahl erforderlich ist, damit ein

Programmdurchlauf ausgewertet wird. Bei einer Adressatenstatistik muss für eine automatische Notengebung überdies noch die Programmnummer und die Notenskala eingegeben werden, für eine Einzelstatistik dagegen die gewünschte Adressatennummer. Bei einer Lehrprogrammstatistik ist natürlich die Angabe der Programmnummer ebenfalls unerlässlich.

Die eingegebenen Auswertekriterien werden von PLAST kontrolliert und nötigenfalls beanstandet. Wenn kein Fehler vorliegt, werden sie zur Kontrolle auf dem gewählten Ausgabemedium ausgedruckt.

Darauf wird das Unterprogramm aktiviert, das die einzelnen Statistikstanzungen der Lehrprogramm-Datensätze einliest und laufend kontrolliert. Fehlerhafte Statistikstanzungen werden sofort beanstandet. Wenn alle Daten eines Programmdurchlaufs eingelesen worden sind, berechnet PLAST die Sortierdaten. Wenn alle diese Sortierdaten innerhalb der angegebenen Schranken liegen, werden die Hauptdaten des ganzen Durchlaufs bestimmt, welche etwa den im Schlussprotokoll enthaltenen Angaben entsprechen.

Bei einer Lehrprogrammstatistik werden darauf die Zähler für die Berechnung der Mittelwerte dieser Hauptdaten nachgeführt. Bei einigen davon werden auch die Extremalwerte bestimmt. Falls eine ausführliche Statistik gewünscht wurde, werden auch die entsprechenden Zähler für die Antwortverteilung nachgeführt. Infolge der beschränkten Grösse des Computerspeichers kann eine ausführliche Lehrprogrammstatistik nur für Lehrprogramme mit höchstens 40 verschiedenen Fragen erstellt werden.

Bei einer Adressatenstatistik wird zunächst geprüft, ob eine automatische Notengebung erforderlich ist. Gegebenenfalls wird aus der erzielten Totalpunktzahl und der eingegebenen Notenskala die Note berechnet und für eine spätere weitere Auswertung zusammen mit der Adressatennummer auf Lochstreifen ausgestanzt. Darauf werden die Hauptdaten des Durchlaufs in kompakter Form auf einer einzigen Zeile ausgedruckt. Falls eine ausführliche Adressatenstatistik verlangt wurde, werden auch die Daten zu jeder einzelnen Antwort (inkl. Resultat bei numerischen Antworten) ausgegeben.

Nachdem alle Daten des Programmdurchlaufs fertig verarbeitet worden sind, wird ein weiterer Datensatz eingelesen. Wenn PLAST das Ende des Lochstreifens feststellt, druckt es auf dem Fernschreiber eine entsprechende Meldung aus und verlangt vom Operateur eine Entscheidung über die Programmfortsetzung. Er kann darauf die Verarbeitung weiterer Lochstreifen einleiten oder den Abbruch der Auswertung und (bei einer Lehrprogrammstatistik) das Ausdrucken der Ergebnisse veranlassen. In den folgenden Kapiteln werden einige Auswertungsmöglichkeiten anhand von Beispielen erläutert.

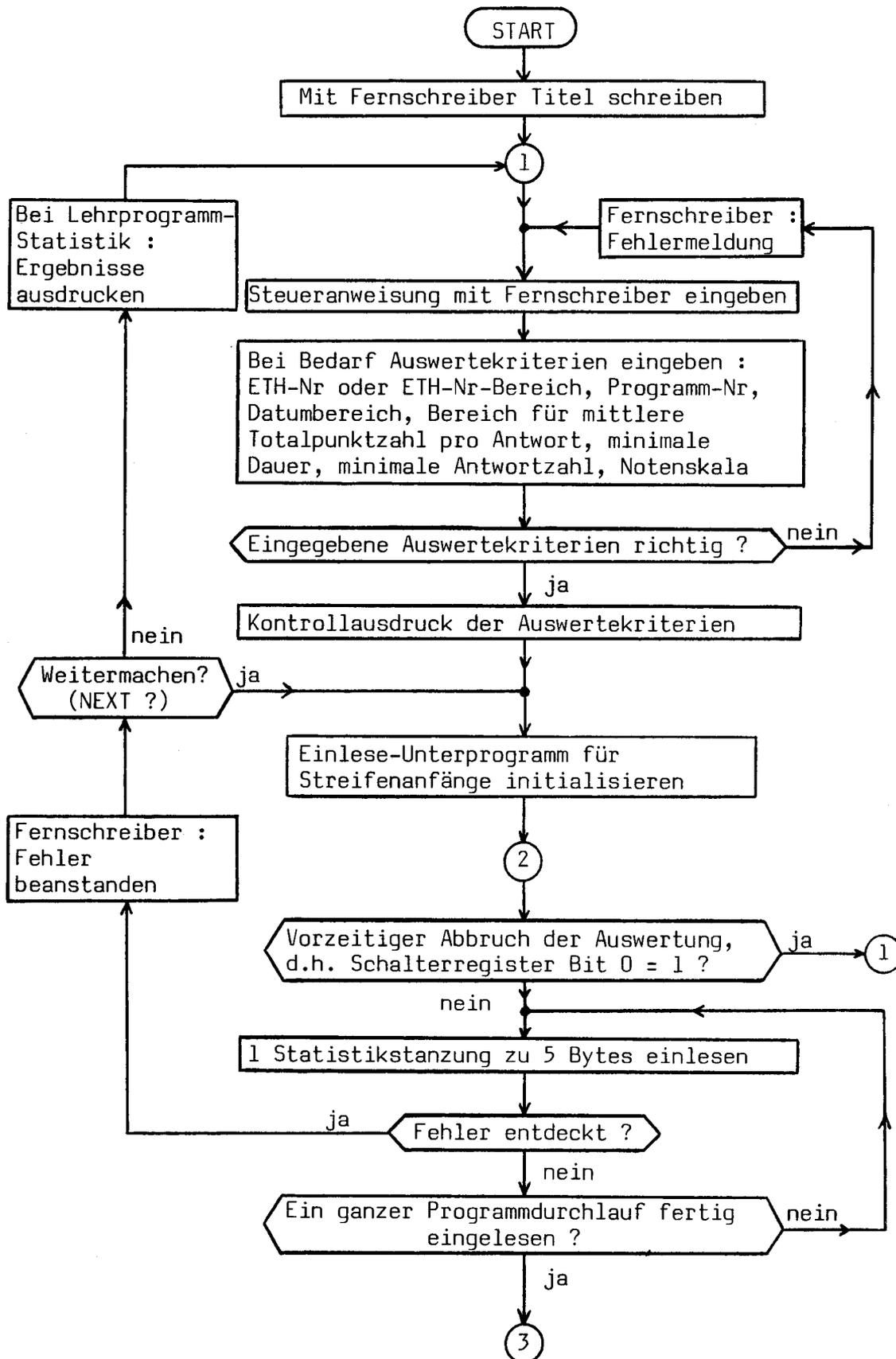


Fig. 7.9a : Grobflussdiagramm des PLANETH-Statistikauswerteprogramms PLAST

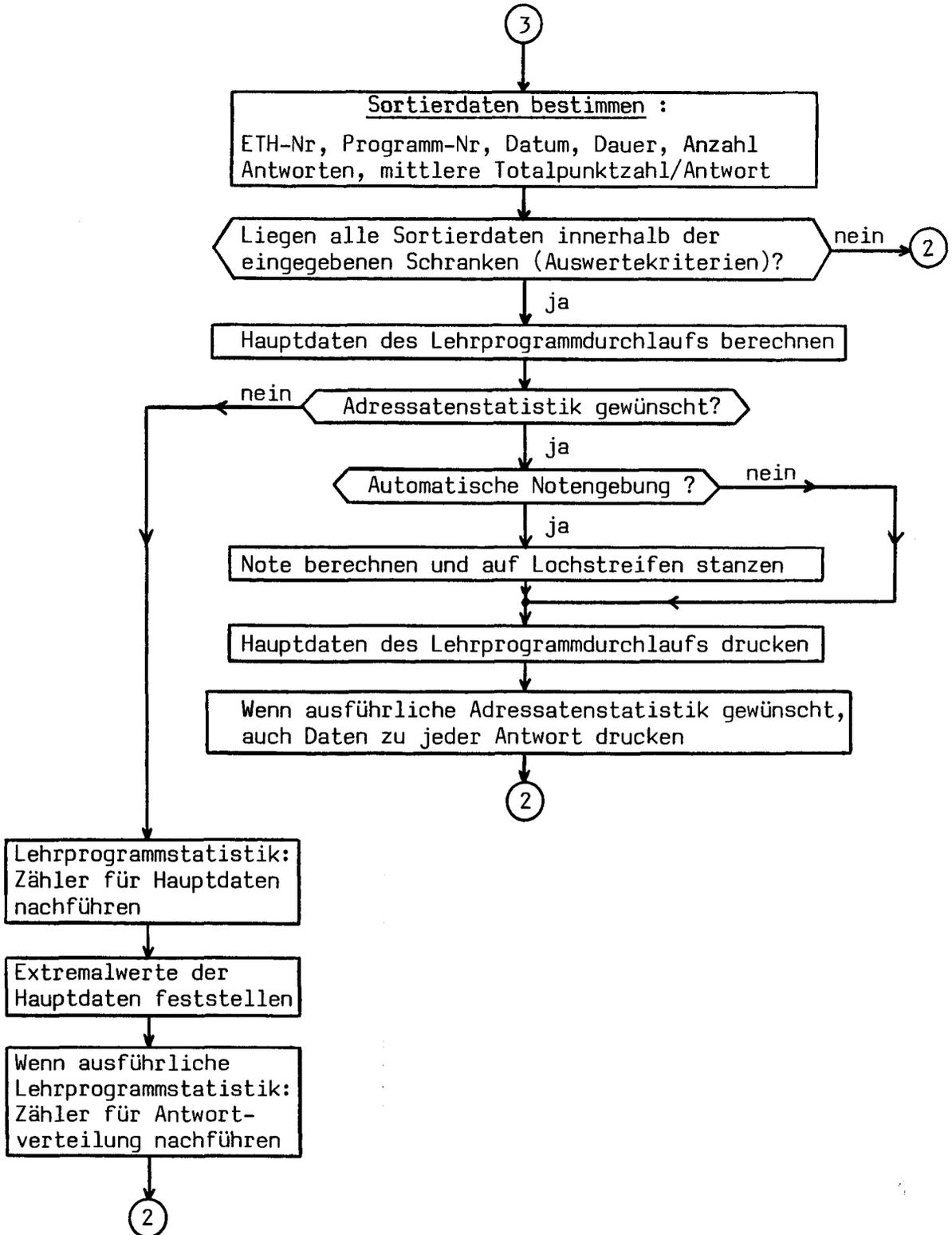


Fig. 7.9b : Grobflussdiagramm des PLANETH-Statistikauswerteprogramms PLAST

7.4.1 Kurze Adressatenstatistik

Eine kurze Adressatenstatistik wird meist erstellt, um einen Ueberblick über den gesamten Lehrbetrieb in einem gewissen Zeitraum zu gewinnen. Fig. 7.10 zeigt einen Ausschnitt aus einer solchen Statistik. Die Darstellung ist sehr kompakt, da alle Hauptdaten eines Programm-durchlaufs auf einer Zeile dargestellt werden müssen.

Zu jeder Statistikauswertung gehört die Angabe der verwendeten Auswertekriterien. Diese Kontrollausgaben sehen alle sehr ähnlich aus.

Ein Beispiel dafür zeigt Fig. 7.11 in Kap. 7.4.2 .

STUDENT	PROG	NOTE	DATUM	ZEIT	DAUER	TOTAL-PTE	ANTW-PTE	ZEIT-PTE	BA	KO	SK
75904896	1114	0	770901	1423	17'	103/ 7.9	103/ 7.9	174/13.4	13	4	0
75907089	1125	0	770901	1456	31'	96/10.6	87/ 9.7	89/ 9.9	9	2	0
75904896	1115	0	770901	1458	34'	75/ 4.2	75/ 4.2	239/13.3	18	4	0
75902767	1114	0	770901	1542	40'	76/ 5.4	76/ 5.4	151/10.8	14	4	0
75907089	1111	0	770901	1552	51'	144/12.0	129/10.7	138/11.5	12	2	0
75908962	1114	0	770901	1602	56'	86/ 6.1	86/ 6.1	141/10.1	14	3	0
75902767	1115	0	770901	1658	74'	95/ 5.0	95/ 5.0	200/10.5	19	4	0
75908962	1115	0	770901	1659	52'	111/ 8.5	111/ 8.5	145/11.2	13	3	0
75904896	1123	0	770905	1427	18'	120/ 6.3	120/ 6.3	249/13.1	19	4	0
75902767	1123	0	770905	1427	26'	134/ 8.4	134/ 8.4	183/11.4	16	1	0
75908962	1123	0	770905	1435	34'	122/ 7.2	122/ 7.2	137/ 8.1	17	3	0
74901497	1111	0	770905	1439	43'	103/ 8.6	92/ 7.7	141/11.7	12	2	0
75904896	1124	0	770905	1441	12'	102/ 7.3	102/ 7.3	186/13.3	14	4	0
75902767	1124	0	770905	1453	24'	115/ 9.6	115/ 9.6	125/10.4	12	1	0
75908962	1124	0	770905	1458	22'	120/12.0	120/12.0	104/10.4	10	3	0
75904896	1125	0	770905	1502	18'	84/ 9.4	75/ 8.3	111/12.3	9	4	0
74901497	1112	0	770905	1522	38'	79/ 6.1	71/ 5.5	151/11.6	13	2	0
75902767	1125	0	770905	1529	28'	50/ 5.6	45/ 5.0	107/11.9	9	1	0
75904896	1111	0	770905	1531	26'	89/ 7.4	78/ 6.5	164/13.7	12	4	0
75908962	1125	0	770905	1531	29'	94/10.5	85/ 9.4	98/10.9	9	3	0
75907089	1112	0	770905	1535	58'	140/12.7	129/11.7	91/ 8.3	11	5	0
75907089	1113	0	770905	1643	67'	124/12.4	115/11.5	77/ 7.7	10	5	0
00000	1121	0	770908	1002	69'	99/ 6.6	99/ 6.6	160/10.7	15	3	0
75908004	1125	0	770908	1417	25'	100/11.1	90/10.0	103/11.4	9	4	0
75903112	1123	0	770908	1431	30'	103/ 5.1	103/ 5.1	246/12.3	20	3	0

Legende :

Bei Punktzahlangaben : Erzielte Punktsumme/Punktmittel pro Antwort

BA = Anzahl bewertete Antworten (für Berechnung des Punktmittels)

KO = Konsolennummer

SK = Stopkriterium (siehe Kap. 7.4.4)

Fig. 7.10 : Ausschnitt aus einer kurzen Adressatenstatistik

7.4.2 Adressatenstatistik mit automatischer Notengebung

In diesem Beispiel werden die Hauptdaten aller Programmdurchläufe des Lehrprogramms llll ausgedruckt. Auf Grund der erzielten Totalpunktzahl wird mit Hilfe einer eingegebenen Notenskala jedem Durchlauf eine Note zugewiesen. Diese Notenskala wird meist anhand einer vorher erstellten Lehrprogrammstatistik festgelegt.

Fig. 7.11 zeigt zunächst die zur Kontrolle ausgedruckten Auswertekriterien, nach denen die Auswertung erfolgte. Man ersieht daraus, dass nur Programmdurchläufe ausgewertet wurden, deren Adressatennummer mit einer Zahl zwischen 68 und 74 begann, die zwischen dem 1. Januar 1975 und dem 28. Februar 1975 absolviert wurden, deren mittlere Totalpunktzahl pro Antwort zwischen 0 und 15,5 lag, die mindestens 5 Minuten dauerten und mindestens 4 Antworten umfassten. Darauf folgt die für die Korrektur verwendete Notenskala. Es sind dabei nur die Knickstellen der Notenfunktion angegeben. Zwischen diesen Punkten wird linear interpoliert.

PLAST 3F

AUSWERTEKRITERIEN:

MODE 0

ETH-NR-JAHRGANG-INTERVALL:

68 74

PROGRAMM:

1111

ANF/ENDDATUM:

750101 750228

MIN/MAX MITTL.TOT-P:

.00 15.50

MIN-DAUER/-ANTWORTZAHL:

5 4

MITTL.TOT-P=NOTE:

.00	10
4.00	20
7.00	40
10.50	50
12.50	60

Fig. 7.11 : Auswertekriterien zur Adressatenstatistik von Fig. 7.12

Fig. 7.12 zeigt einen Ausschnitt aus der Adressatenstatistik mit automatischer Notengebung, die sich mit den Auswertekriterien gemäss Fig. 7.11 ergibt. Der angegebene Notenwert ist eine ganze Zahl mit dem zehnfachen Wert der effektiv erzielten Note. Dieser Wert wird bei der automatischen Notengebung zusammen mit der Adressatennummer für spätere weitere Auswertungen auf Lochstreifen ausgestanzt. Die Bedeutung der verwendeten Abkürzungen wurde bereits in Fig. 7.10 erklärt.

STUDENT	PROG	NOTE	DATUM	ZEIT	DAUER	TOTAL-PTE	ANTW-PTE	ZEIT-PTE	BA	KO	SK
73901274	1111	25	750110	1255	38'	43/ 4.7	39/ 4.3	86/ 9.6	9	1	1
73910358	1111	36	750110	1556	35'	57/ 6.4	52/ 5.8	92/10.2	9	3	1
73903528	1111	48	750117	1240	30'	88/ 9.7	79/ 8.8	99/11.0	9	3	1
72907900	1111	35	750117	1321	37'	57/ 6.3	52/ 5.8	83/ 9.2	9	3	0
72910573	1111	48	750117	1339	18'	89/ 9.9	79/ 8.8	116/12.9	9	3	0
73911174	1111	37	750117	1416	15'	59/ 6.5	52/ 5.8	120/13.3	9	4	1
72902030	1111	26	750117	1453	10'	44/ 4.9	39/ 4.3	125/13.9	9	4	0
72901267	1111	60	750117	1503	8'	120/13.3	105/11.7	127/14.1	9	4	0
73903676	1111	56	750117	1713	10'	105/11.7	92/10.2	132/14.7	9	3	1
73905911	1111	60	750123	1229	26'	118/13.1	105/11.7	111/12.3	9	3	1
73905895	1111	60	750123	1229	26'	117/13.0	105/11.7	106/11.8	9	4	0
73903296	1111	36	750123	1230	25'	58/ 6.4	52/ 5.8	98/10.9	9	2	1
73903296	1111	60	750123	1236	6'	120/13.4	105/11.7	130/14.4	9	2	1
73913634	1111	18	750123	1332	23'	29/ 3.2	26/ 2.9	105/11.7	9	4	1
73910937	1111	43	750123	1527	31'	72/ 8.0	65/ 7.2	96/10.7	9	2	0
73901738	1111	35	750123	1726	37'	57/ 6.3	52/ 5.8	83/ 9.2	9	2	0
73908014	1111	25	750123	1741	36'	43/ 4.7	39/ 4.3	84/ 9.3	9	4	1
72902034	1111	60	750124	1514	7'	120/13.3	105/11.7	128/14.2	9	4	0
72901267	1111	60	750124	1523	16'	132/14.7	118/13.1	110/12.2	9	3	0
73906455	1111	26	750124	1534	18'	44/ 4.9	39/ 4.3	117/13.0	9	4	1
71913560	1111	10	750124	1701	5'	0/ .0	0/ .0	85/14.2	6	4	1
73906737	1111	43	750124	1706	34'	72/ 8.0	65/ 7.2	92/10.2	9	2	1
73900151	1111	37	750130	1239	26'	59/ 6.6	53/ 5.9	105/11.7	9	1	0
73900151	1111	60	750130	1251	7'	119/13.2	104/11.6	127/14.1	9	1	0
73905820	1111	60	750130	1332	24'	133/14.7	118/13.1	112/12.4	9	1	1
73903536	1111	60	750130	1337	27'	132/14.6	118/13.1	104/11.6	9	4	1
73900755	1111	36	750130	1531	29'	57/ 6.4	52/ 5.8	94/10.4	9	4	1
72906035	1111	43	750130	1623	35'	71/ 7.9	65/ 7.2	86/ 9.6	9	3	1
73908345	1111	47	750130	1657	34'	86/ 9.6	79/ 8.8	84/ 9.3	9	3	0
73902181	1111	18	750130	1745	54'	28/ 3.1	26/ 2.9	66/ 7.3	9	1	1
72909013	1111	43	750131	1239	29'	73/ 8.1	66/ 7.3	99/11.0	9	1	1
73907826	1111	43	750131	1257	37'	71/ 7.9	65/ 7.2	88/ 9.8	9	3	0
72910573	1111	48	750131	1318	12'	58/ 9.7	52/ 8.7	70/11.7	6	1	1
72904485	1111	55	750131	1319	15'	103/11.4	91/10.1	117/13.0	9	4	0

Fig. 7.12 : Ausschnitt aus Adressatenstatistik mit autom. Notengebung

7.4.3 Ausführliche Adressatenstatistik

Bei der ausführlichen Adressatenstatistik werden bei jedem Programmdurchlauf neben den Hauptdaten auch die Daten der einzelnen Antworten ausgedruckt. Auf diese Weise erhält der Lehrprogrammautor ein lückenloses Protokoll des Lernprozesses. Solche ausführlichen Adressatenstatistiken sind besonders nützlich beim ersten Einsatz neuer Lehrprogramme, da irreguläre Programmabläufe sofort erkennbar sind. Sie erlauben auch eine detaillierte Analyse des Lernverhaltens einzelner Adressaten bei mehreren verschiedenen Lehrprogrammen.

STUDENT	PROG	NOTE	DATUM	ZEIT	DAUER	TOTAL-PTE	ANTW-PTE	ANTW-PTE	ANTW-PTE	BA	KO	SK
73912479	1114	0	770303	1539	16'	42/ 5.2	42/ 5.2	99/12.4	8	2	1	
NR	PRZ	BEW	REP	ANTWORT	ANTW-PTE	ZEIT-PTE	ZEITVERBR	WERT				
1	87	1	0	2	8	13	20 S	.0000E+00				
2	93	1	0	4	0	11	30 S	.0000E+00				
3	93	1	1	3	7	14	10 S	.0000E+00				
4	108	1	0	1	2	11	60 S	.0000E+00				
5	108	1	1	2	6	14	20 S	.0000E+00				
6	120	1	0	3	10	12	50 S	.0000E+00				
7	150	1	0	4	9	11	60 S	.0000E+00				
8	160	1	0	M4	0	13	110 S	-.2400E-04				

STUDENT	PROG	NOTE	DATUM	ZEIT	DAUER	TOTAL-PTE	ANTW-PTE	ANTW-PTE	ANTW-PTE	BA	KO	SK
74909599	1111	0	770303	1720	48'	61/ 5.1	54/ 4.5	149/12.4	12	3	0	
NR	PRZ	BEW	REP	ANTWORT	ANTW-PTE	ZEIT-PTE	ZEITVERBR	WERT				
1	9	1	0	3	0	13	50 S	.0000E+00				
2	28	1	0	2	0	12	100 S	.0000E+00				
3	44	1	0	3	13	13	80 S	.0000E+00				
4	63	1	0	5	11	15	20 S	.0000E+00				
5	82	1	0	3	13	6	220 S	.0000E+00				
6	101	1	0	5	2	12	90 S	.0000E+00				
7	117	1	0	3	0	14	30 S	.0000E+00				
8	136	1	0	4	1	12	130 S	.0000E+00				
9	262	0	0	W	0	0	130 S	.0000E+00				
10	262	0	1	P4	0	0	30 S	.9900E+02				
11	155	1	0	P4	0	11	320 S	.9000E+01				
12	173	1	0	F0	14	11	530 S	.5000E+00				
13	182	1	0	P4	0	15	20 S	.6250E+00				
14	191	1	0	P4	0	15	20 S	.2500E-01				
15	230	0	0	5	0	0	00 S	.0000E+00				
16	232	0	0	2	0	0	20 S	.0000E+00				
17	234	0	0	4	0	0	10 S	.0000E+00				
18	236	0	0	1	0	0	10 S	.0000E+00				
19	238	0	0	2	0	0	20 S	.0000E+00				
20	240	0	0	1	0	0	10 S	.0000E+00				

Fig. 7.13 : Ausschnitt aus einer ausführlichen Adressatenstatistik

Fig. 7.13 zeigt einen Ausschnitt aus einer solchen ausführlichen Adressatenstatistik. Die für die Ausgabe der Hauptdaten verwendeten Abkürzungen wurden schon in Fig. 7.10 erklärt. Zusätzlich kommen noch folgende Abkürzungen vor :

PRZ = Zeilennummer des zur entsprechenden Frage gehörenden BILD-Befehls

BEW = Bewertung : 1 bedeutet bewertete, 0 unbewertete Antwort

REP = Repetition: 1 bedeutet Repetition, 0 Erstantwort.

WERT = Eingegebenes Resultat bei numerischen Antworten

Unter ANTWORT ist bei Auswahlantworten die Nummer der gedrückten Taste, bei numerischen Antworten dagegen die Fehlerklasse (siehe Fig. 6.12) angegeben. W bedeutet WEITER-Taste, Z Zeitablauf.

7.4.4 Kurze Lehrprogrammstatistik

Bei der kurzen Lehrprogrammstatistik werden aus den Hauptdaten der Durchläufe eines bestimmten Lehrprogramms die Mittelwerte, Maxima und Minima berechnet. Daneben werden die prozentuale Verteilung der mittleren Totalpunktzahl pro Antwort und die prozentuale Verteilung der Stopkriterien (Definition siehe übernächste Seite) bestimmt. Diese Angaben erlauben dem Programmautor bereits eine gute Abschätzung des Schwierigkeitsgrades des ganzen Lehrprogramms und des Lernerfolgs der Adressaten.

Fig. 7.14 zeigt eine solche kurze Lehrprogrammstatistik für das Lehrprogramm 1113. MODE 0 bedeutet, dass bei jeder Frage Erstantworten und Repetitionen, also alle Antworten, berücksichtigt werden. Neben den Mittelwerten werden in Klammern die Maximal- und Minimalwerte angegeben.

Bei der Verteilung der mittleren Totalpunktzahl \bar{p}_t pro Antwort ist jeweils die prozentuale Anzahl Programmdurchläufe angegeben, welche die Bedingung $n \leq \bar{p}_t < n+1$ erfüllen, wobei n zwischen 0 und 25 liegt.

Bei der anschliessenden Verteilung der Stopkriterien wird die prozentuale Anzahl Programmdurchläufe angegeben, die auf ein bestimmtes Kriterium entfallen.

LEHRPROGRAMM 1113

MODE 0
PROGRAMMZAHL 254
STUDENTEN-STD 205.0

RESULTATE : MITTELWERT (MAX/MIN) , PRO ANTWORT (MAX./MIN.)

ANTWORTZAHL: 12.19 (17/ 4)
DAUER (MIN): 48.42 (105/ 5) , 3.97
TOTALPUNKTE: 71.43 (149/ 0) , 5.86 (14.9/ .0)
ANTW.PUNKTE: 64.21 (130/ 0) , 5.27 (13.0/ .0)
ZEIT-PUNKTE: 140.65 (225/ 39) , 11.54 (15.0/ 6.5)

VERTEILUNG DER TOTALPUNKTE PRO ANTWORT :

BEREICH		VERTEILUNGSFUNKTION
0 - 1	4.72 %	4.72 %
1 - 2	4.72 %	9.45 %
2 - 3	9.06 %	18.50 %
3 - 4	11.81 %	30.31 %
4 - 5	11.42 %	41.73 %
5 - 6	15.35 %	57.09 %
6 - 7	8.27 %	65.35 %
7 - 8	7.48 %	72.83 %
8 - 9	8.27 %	81.10 %
9 - 10	7.87 %	88.98 %
10 - 11	2.76 %	91.73 %
11 - 12	2.36 %	94.09 %
12 - 13	3.54 %	97.64 %
13 - 14	.79 %	98.43 %
14 - 15	1.57 %	100.00 %
15 - 16	.00 %	100.00 %
16 - 17	.00 %	100.00 %
17 - 18	.00 %	100.00 %
18 - 19	.00 %	100.00 %
19 - 20	.00 %	100.00 %
20 - 21	.00 %	100.00 %
21 - 22	.00 %	100.00 %
22 - 23	.00 %	100.00 %
23 - 24	.00 %	100.00 %
24 - 25	.00 %	100.00 %
25 - 26	.00 %	100.00 %

STOPKRITERIEN:

0 78.74 %
1 21.26 %
2 .00 %
3 .00 %
4 .00 %
5 .00 %

Fig. 7.14 : Kurze Lehrprogrammstatistik für das Programm 1113

Diese Stopkriterien geben den Grund für den Abschluss eines Lehrprogrammdurchlaufs an und haben folgende Bedeutung :

- 0 = Programmende (Unbedingter STOP-Befehl)
- 1 = STOP-Taste gedrückt
- 2 = Untere Punktlimite unterschritten (in bedingtem STOP-Befehl)
- 3 = Obere Punktlimite überschritten (in bedingtem STOP-Befehl)
- 4 = Systemstop (Zwangstop vor Abschalten des Lehrsystems)
- 5 = Notstop wegen zu vieler Statistikstanzungen.

7.4.5 Ausführliche Lehrprogrammstatistik

Bei einer ausführlichen Lehrprogrammstatistik wird neben einer kurzen Lehrprogrammstatistik eine Antwortstatistik ausgedruckt, in der zu jeder Frage folgende Angaben zu finden sind :

- Fragenummer und Programm-Zeilenummer des entsprechenden BILD-Befehls
- Mittlere verbrauchte Zeit für die Beantwortung
- Mittlere erzielte Antwortpunktzahl
- Mittlere Zeitpunktzahl
- Absolute und prozentuale Verteilung der Antworten. Bei Fragen mit Auswahlantworten ist die Verteilung auf die 5 Tasten 1 - 5, WEITER und Zeitablauf angegeben. Bei Fragen mit numerischen Konstruktivantworten ist die Verteilung auf die 11 Fehlerklassen gemäss Fig. 6.12, WEITER und Zeitablauf angegeben.

Fig. 7.15 zeigt einen Ausschnitt aus einer solchen Antwortstatistik. Die Fragen 1 und 4 besitzen Auswahlantworten, die Fragen 2 und 3 dagegen numerische Antworten. Die zugehörige kurze Lehrprogrammstatistik, die zusammen mit dieser Antwortstatistik die ausführliche Lehrprogrammstatistik über das betreffende Lehrprogramm bildet, ist in Fig. 7.14 zu finden.

Die kurze Lehrprogrammstatistik ermöglicht eine Beurteilung des Lehrprogramms als Ganzes. Die Angaben der Antwortstatistik geben dagegen Aufschluss über die Qualität und den Schwierigkeitsgrad der einzelnen Fragelehre-schritte.

FRAGE 1 , ZEILE 6

 MITTELWERTE: ZEIT 184 S ,ANTW-P 4.73 , ZEIT-P 11.40
 ANTWORTEN : 375 = 100.0 %
 1 78 = 20.8 %
 2 149 = 39.7 %
 3 51 = 13.6 %
 4 29 = 7.7 %
 5 54 = 14.4 %
 W 14 = 3.7 %
 Z 0 = .0 %

FRAGE 2 , ZEILE 15

 MITTELWERTE: ZEIT 421 S ,ANTW-P 4.10 , ZEIT-P 10.69
 ANTWORTEN : 242 = 100.0 %
 M4 11 = 4.5 %
 M3 90 = 37.2 %
 M2 0 = .0 %
 M1 0 = .0 %
 F0 59 = 24.4 %
 P1 0 = .0 %
 P2 0 = .0 %
 P3 6 = 2.5 %
 P4 9 = 3.7 %
 K 25 = 10.3 %
 V 3 = 1.2 %
 W 37 = 15.3 %
 Z 2 = .8 %

FRAGE 3 , ZEILE 20

 MITTELWERTE: ZEIT 256 S ,ANTW-P 4.87 , ZEIT-P 12.06
 ANTWORTEN : 325 = 100.0 %
 M4 16 = 4.9 %
 M3 18 = 5.5 %
 M2 0 = .0 %
 M1 0 = .0 %
 F0 109 = 33.5 %
 P1 0 = .0 %
 P2 0 = .0 %
 P3 0 = .0 %
 P4 103 = 31.7 %
 K 22 = 6.8 %
 V 1 = .3 %
 W 55 = 16.9 %
 Z 1 = .3 %

FRAGE 4 , ZEILE 35

 MITTELWERTE: ZEIT 112 S ,ANTW-P 5.02 , ZEIT-P 12.05
 ANTWORTEN : 246 = 100.0 %
 1 37 = 15.0 %
 2 65 = 26.4 %
 3 21 = 8.5 %
 4 17 = 6.9 %
 5 103 = 41.9 %
 W 3 = 1.2 %
 Z 0 = .0 %

Fig. 7.15 : Ausschnitt aus einer Antwortstatistik

Betrachtet man die mittlere bei einer Frage erzielte Antwortpunktzahl im Verhältnis zur erreichbaren Punktzahl für die richtige Antwort, so sieht man sofort, wie schwierig diese Frage ist.

Die mittlere Zeitpunktzahl sagt dagegen aus, ob die Vorgabezeit für diese Frage zu lang, richtig oder zu kurz ist. Nach den bisher gemachten Erfahrungen sollte die mittlere Zeitpunktzahl bei jeder Frage zwischen etwa 11 und 13 liegen. Beträgt sie weniger als 10 Punkte, so muss die Vorgabezeit in der Regel vergrößert werden, da sonst die Zeitabläufe (Z) beträchtlich ansteigen und viele Adressaten bei der Bearbeitung dieser Frage unter Zeitdruck stehen, was sich negativ auf die Lernmotivation auswirkt. Liegt die mittlere Zeitpunktzahl hingegen über 14, so kann die Vorgabezeit etwas verkleinert werden.

Die Verteilung der Antworten auf eine Frage zeigt, wie oft die einzelnen Auswahlantworten gewählt wurden resp. wie häufig das eingegebene Resultat in einer bestimmten Fehlerklasse lag und wie oft freiwillig mit der WEITER-Taste (W) oder unfreiwillig infolge Zeitablaufs (Z) auf die Beantwortung der Frage verzichtet wurde.

Auf Grund dieser Antwortverteilung sieht man, ob der betreffende Fragelehrschritt verbessert werden sollte. Vielleicht ist die Vorinformation nicht ausreichend oder die Fragestellung unklar. Bei Fragen mit Auswahlantworten kann man nach häufig gewählten Falschantworten zusätzliche Erklärungen einbauen und nur selten gegebene Falschantworten durch bessere ersetzen. Bei Fragen mit numerischen Konstruktivantworten können bei oft vorkommenden Fehlerklassen ebenfalls zusätzliche Erläuterungen vorgesehen werden. Bei abnormer Häufigkeit einer bestimmten Fehlerklasse (z.B. M3 bei Frage 2, P4 bei Frage 3 in Fig. 7.15) kann man mit einer anschliessenden ausführlichen Adressatenstatistik feststellen, ob die eingegebenen Resultate einen Häufungspunkt besitzen. Bei P4 in Frage 3 hat beispielsweise die Mehrzahl der eingegebenen falschen Resultate einen ganz bestimmten Wert, der einem typischen Fehler entspricht. Auch bei solchen typischen Fehlern ist die Verwendung von zusätzlichen Hinweisen und Erklärungen angezeigt.

8. BETRIEBSERFAHRUNGEN MIT DEM LEHRSYSTEM PLANETH

Das Lehrsystem PLANETH wird von den Studenten der Kurse "Technische Elektrizitätslehre III" im 3. Semester und "Verteilte Netzwerke" im 6. Semester auf freiwilliger Basis rege benützt. Von September 1974 bis März 1978 wurden 5331 statistisch erfasste Programmdurchläufe in etwa 3200 Studentenkontaktstunden absolviert. Die Beteiligung der Studenten nimmt dabei ständig zu : Im letzten Jahr, vom April 1977 bis März 1978, wurden nicht weniger als 1829 Programmdurchläufe registriert. Gegenwärtig sind 15 PLANETH-Lehrprogramme verfügbar (5 davon mit Ton), 3 weitere sind noch in Entwicklung. Fig. 8.1 zeigt das Lehrsystem PLANETH im Betrieb.



Fig. 8.1 : PLANETH-Lehrkonsolen im Betrieb

8.1 PLANETH im Urteil der Studenten

Um eine Aussage darüber zu gewinnen, wie die PLANETH-Benützer das Lehrsystem und einige seiner prinzipiellen Möglichkeiten beurteilen, wurde bei 4 Lehrprogrammen am Schluss jedes Durchlaufs eine kleine Umfrage durchgeführt, die zusammen mit den übrigen Statistikdaten ausgewertet wurde.

Die am meisten interessierende Frage ist natürlich das Urteil der Studenten über das Lehrsystem PLANETH selbst. In der Zeit vom März 1975 bis Februar 1976 beurteilten von 529 Studenten ca. 63% PLANETH sehr positiv, 20% äuserten

keine Meinung und ca. 17% gaben ein eher negatives Urteil ab (siehe [52]). Daraufhin wurden sowohl am PLANETH-Systemprogramm als auch an den betreffenden Lehrprogrammen verschiedene Verbesserungen vorgenommen, die zur Folge hatten, dass die Beurteilung von PLANETH heute noch viel günstiger ausfällt.

In der Zeit von November 1976 bis Februar 1978 wurde bei 1241 Programmdurchläufen wiederum die gleiche Frage gestellt. Dabei beurteilten nun im Mittel 77,3% PLANETH positiv, 9,7% äusserten keine Meinung und nur noch 13,0% gaben ein eher negatives Urteil ab. Die genaue Fragestellung, die entsprechenden Auswahlantworten und die Verteilung der Antworten auf diese sind in Fig. 8.2 angegeben.

Frage : Wie empfinden Sie diese Uebungsmethode
(Uebungen auf dem Lehrsystem PLANETH) ?

<u>Antworten</u> :	Total	:	1241	≐	100,0%
-ich halte diese Methode für sehr gut		:	604	≐	48,7%
-ich halte sie für etwa gleichwertig mit dem Lösen von "Hausaufgaben"		}	355	≐	28,6%
-diese Methode ist langweilig		:	45	≐	3,6%
-man fühlt sich unangenehm unter Druck und ausgefragt		:	81	≐	6,5%
-es ist eine Drillmethode, auf die man an einer Hochschule verzichten sollte		}	36	≐	2,9%
-keine Antwort		:	120	≐	9,7%

Fig. 8.2 : Beurteilung von PLANETH durch die Benützer (Nov.76-Feb.78)

Es ist sehr interessant, dass der Prozentsatz der positiven Beurteilungen recht stark vom Lehrprogramm abhängt, in dem nach dem Urteil über PLANETH gefragt wurde. In der erwähnten Beobachtungsperiode schwankte dieser Prozentsatz je nach Lehrprogramm zwischen 82,4% und 72,3%. Auffällig ist auch, dass gute Studenten PLANETH gegenüber positiver eingestellt sind als schlechte Studenten.

Zusammenfassend kann man also feststellen, dass das Lehrsystem PLANETH bei den Benützern sehr beliebt ist, wobei die positiven und negativen Urteile etwa im Verhältnis 6 : 1 stehen.

Weiter interessierte die Frage, wie die Benutzer Lehrprogramme mit numerischen Freiwahlantworten beurteilen, die bei PLANETH mit sehr geringem Aufwand realisierbar sind. Von November 1976 bis Februar 1978 beurteilten von 853 Benutzern 52,3% Fragen mit numerischen Konstruktivantworten sehr positiv. Weitere 19,3% hielten solche Fragen für gleichwertig mit Auswahlfragen, während 19,0% ein eher negatives Urteil abgaben und 9,4% keine Meinung äusserten. Von den 19% mit negativen Meinungen fanden gegen 11%, dass solche Fragen zwar lehrreicher seien, wobei die numerische Eingabe jedoch etwas mühsam sei. Diese Ansicht dürfte teilweise darauf zurückzuführen sein, dass die Eingabe nicht genau gleich wie bei den heute gebräuchlichen Taschenrechnern erfolgte, so dass beim Adressaten jeweils ein gewisses Umdenken nötig war. Dieser offenbar etwas störende Unterschied wurde inzwischen durch eine entsprechende Aenderung des PLANETH-Systemprogramms beseitigt.

Numerische Konstruktivantworten werden noch etwas positiver beurteilt, wenn sie in einem vertonten Lehrprogramm vorkommen (57,1% positive zu 18,7% negative Meinungen). Allerdings ist dann natürlich auch der Aufwand für die Programmherstellung beträchtlich grösser. Durch Fragen mit numerischen Konstruktivantworten statt nur mit Auswahlantworten lassen sich aber auch unvertonte Programme mit wenig zusätzlichem Aufwand wesentlich interessanter und lehrreicher gestalten.

Die Herstellung von vertonten Lehrprogrammen ist wesentlich aufwendiger als die von unvertonten Programmen. Es stellte sich deshalb die Frage, wie die Benutzer diese Vertonung beurteilen. In der Beobachtungsperiode vom November 1976 bis Februar 1978 hielten von 332 befragten Studenten 49,4% vertonte Lehrprogramme für lehrreicher und interessanter, 19,3% betrachteten sie als gleichwertig mit unvertonten Programmen, 23,2% empfanden den Ton als manchmal eher störend und 8,1% äusserten keine Meinung.

Bei dem grossen Aufwand für die Vertonung überrascht der relativ hohe Anteil an negativen Meinungen doch etwas. Andererseits wird das Lehrsystem PLANETH insgesamt in vertonten Programmen mit numerischen Konstruktivantworten am günstigsten beurteilt. Lehrprogramme, die alle Möglichkeiten von PLANETH ausschöpfen, scheinen also sehr grossen Anklang zu finden.

Die oben erwähnte Untersuchung über die Beliebtheit vertonter Programme wurde bei einem einzigen Lehrprogramm durchgeführt. Falls also die Vertonung bei diesem Programm etwas ungeschickt durchgeführt wurde, schlägt sich dies naturgemäss voll in der Beurteilung nieder. Man muss dieses Resultat also eher vorsichtig interpretieren und mit Verallgemeinerungen zurückhaltend sein.

Auch über die von den Benützern als optimal empfundene Dauer eines Lehrprogrammdurchlaufs wurden eingehende Erhebungen durchgeführt. Es zeigte sich, dass Programme mit einer mittleren Bearbeitungsdauer von unter 30 Minuten überwiegend als zu kurz angesehen werden, während solche von über 60 Minuten Dauer als zu lang betrachtet werden. Die richtige Lehrprogrammdauer liegt offensichtlich bei etwa 45 Minuten, denn solche Programme werden mehrheitlich als "in der Länge richtig" beurteilt und die Meinungen "zu kurz" und "zu lang" halten sich etwa die Waage.

Die optimale Programmdauer von 45 Minuten erleichtert auch die Schaffung eines Reservationssystems, das wegen des zeitweise sehr grossen Andrangs und der beschränkten Anzahl Lehrkonsolen (gegenwärtig 6) unbedingt notwendig ist. Für jeden Studenten wird pro Lehrprogramm eine Stunde reserviert, was trotz der individuellen Streuung der Bearbeitungszeit praktisch immer für die ungestörte Absolvierung eines Lehrprogramms ausreicht. Bei den 15 gegenwärtig im Einsatz stehenden PLANETH-Lehrprogrammen beträgt die mittlere Bearbeitungszeit im Durchschnitt etwa 35 Minuten und liegt damit noch etwas unter dem Optimalwert.

8.2 Die Zuverlässigkeit des Lehrsystems PLANETH

Wie bei jedem technischen System können auch bei PLANETH ab und zu kleinere Störungen auftreten. Bei den meisten dieser Störungen handelt es sich um harmlose mechanische Transportstörungen der Bild- und Tonspeicher, die immer nur eine einzige Konsole betreffen.

Bei den Diaprojektoren kann gelegentlich das eingelegte Magazin blockieren. Diese Blockierstörungen sind meist auf unsorgfältiges Einlegen der Magazine zurückzuführen. Es kommt auch vor, dass sich das Zahnrad für die Magazinpositionierung bei den verwendeten Kodak-S-RA-Anwahlprojektoren allmählich etwas verschiebt. In diesem Fall muss jeweils die Nullage dieses Zahnrades wieder richtig adjustiert werden. Solche Magazinblockierstörungen sind jedoch bei sorgfältigem Magazinwechsel sehr selten: Pro 100 Lehrprogrammdurchläufe tritt höchstens eine solche Magazinstörung auf.

Früher fiel bei den verwendeten Projektoren etwa alle 100-200 Betriebsstunden die 150W-Halogenlampe aus, wodurch mitten im Lehrbetrieb lästige Auswechselarbeiten erforderlich wurden. Seit Juli 1976

werden deshalb neue 100W-Lampen eingesetzt, die eine etwas niedrigere Farbtemperatur (etwas gelberes Licht) haben als die Originallampen, dafür aber eine mittlere Lebensdauer von 2000 Stunden aufweisen. Die Bildhelligkeit ist dabei immer noch bei weitem ausreichend. Seither musste keine einzige Projektionslampe mehr ausgewechselt werden. Als weitere positive Nebenwirkungen dieser Massnahme sind die Energieeinsparung und die Erleichterung der Projektorkühlung infolge der geringeren anfallenden Verlustleistung zu erwähnen.

Auch die verwendeten PLANETH-Kassettentonbandgeräte (siehe Kap. 3.10 und 3.11) arbeiten sehr zuverlässig. Zählfehler, die das Aufsuchen einer falschen Tonsequenz bewirken, kommen praktisch nicht vor. Von Oktober 1976 bis März 1978 wurde bei über 25000 ausgezählten Sequenzmarkierimpulsen nur ein einziger Zählfehler registriert, die Hilfstaste "TONSTOERUNG" muss also praktisch nie benützt werden. Dagegen erweisen sich die verwendeten Scotch AVC-60 Kassetten nach längerem Lagern gelegentlich als sehr schwergängig, so dass die Kraft der Antriebsmotoren für einen einwandfreien Bandtransport nicht mehr ausreicht. Abhilfe bringt ein kurzes Schütteln lange nicht benützter Kassetten vor dem Einschieben ins Tonbandgerät. Die Ursache dieses Effektes konnte bis heute nicht genau abgeklärt werden. Andere Fabrikate zeigen diesen Fehler nicht, haben dafür aber andere, ebenfalls störende Nebeneffekte.

Die Elektronik der PLANETH-Lehrkonsolen wurde für möglichst grosse Zuverlässigkeit ausgelegt (siehe Kap. 3). Die geringe Zahl der in der Zeit vom September 1974 bis März 1978 beobachteten Störungen zeigt, dass dieses Ziel erreicht wurde. Es mussten nur drei ernsthaftere Konsolendefekte repariert werden: Einmal machte sich eine kalte Lötstelle bemerkbar, einmal war ein Operationsverstärker defekt und nach über dreijährigem Betrieb fiel eine Stelle einer numerischen Anzeigeeinheit aus. Daneben musste ab und zu eine Tastenlampe oder eine infolge eines zu hohen Einschaltstromstosses (viele Trafos in Konsole) durchgebrannte Konsolenhauptsicherung ersetzt werden.

Dank der grossen Zuverlässigkeit des Computers, der zugehörigen Peripheriegeräte und des Vorschaltgerätes sind Störungen, die das ganze Lehrsystem lahmlegen, äusserst selten. In der Zeit von September 1974 bis März 1978 trat nur eine schwerwiegende Störung auf: Im Dezember 1975 war der PLANETH-Lehrbetrieb wegen eines defekten Computer-Rechenprints für einige Tage unterbrochen. Da der verwendete Computer über keine automatische Abschaltung und Wiedereinschaltung bei Netzausfällen verfügt, kam es einige Male bei kurzzeitigen Netzunterbrüchen (z.B. während Gewittern) zu kleineren Störungen. Da inzwischen für das Lehrsystem PLANETH ein neuer Computer bestellt wurde, der über diese Netzausfallsautomatik verfügt, werden diese Netzunterbruchsstörungen bald der Vergangenheit angehören.

Auch das verwendete PLANETH-Systemprogramm arbeitet sehr zuverlässig. Dank der noch überschaubaren Grösse kommen nach dem gründlichen Ausprüfen der jeweiligen Version keine spontanen Systemzusammenbrüche vor.

Zusammenfassend kann man also festhalten, dass sich die Zuverlässigkeit des Lehrsystems PLANETH in über 3½-jährigem Betrieb als sehr gross erwiesen hat. PLANETH eignet sich somit auch unter diesem Gesichtspunkt ausgezeichnet für den praktischen CUU-Einsatz.

8.3 Die Kosten beim Lehrsystem PLANETH

In diesem Kapitel sollen die effektiven Kosten k_{CUU} pro CUU-Stunde gemäss Formel (1.1) für das Lehrsystem PLANETH berechnet werden. Zur Vereinfachung der Rechnung sei angenommen, dass ein Lehrprogrammdurchlauf zusammen mit der bei einem Reservationssystem unvermeidlichen Pufferzeit bis zum Eintreffen des nächsten Konsolenbenützers gerade eine Stunde dauert, d.h. 1 Lehrprogrammdurchlauf entspricht 1 Stunde CUU. Die Amortisationsdauer d_A für die Hardware des Lehrsystems wird dabei für die folgenden Berechnungen zu 5 Jahren angenommen.

8.3.1 Die Kosten beim realisierten 6-Konsolen-System

Der verwendete Computer und die zugehörigen Standardperipheriegeräte waren schon etwa 5 Jahre im Einsatz, als sie 1974 fest dem PLANETH-System zugeteilt wurden; sie sind also bereits vollständig abgeschrieben. Deshalb muss nur eine speziell für das PLANETH-System vorgenommene Modifikation der Fernschreibmaschine (automatische Ein- und Ausschaltung), die ca. 2000 Fr kostete, für die Berechnung der effektiven Hardware-Investitionskosten I_H berücksichtigt werden. Somit ergeben sich folgende Posten für die Berechnung von I_H :

Computer inkl. Standardperipherie	0 Fr	(abgeschrieben)
Umbau Fernschreibmaschine	2000 Fr	
Vorschaltgerät (siehe Kap. 4.3)	10000 Fr	
6 PLANETH-Konsolen	<u>123000 Fr</u>	

$$\text{Totale Hardware-Investitionskosten } I_{H_{P6}} = 135000 \text{ Fr} \quad (8.1)$$

In den angegebenen Werten sind nicht nur Materialkosten, sondern auch einigermaßen realistische Arbeitskosten für die Herstellung des Vorschaltgerätes und der Lehrkonsolen enthalten, jedoch keine Entwicklungskosten. Bei einer Abschreibungsdauer $d_A = 5$ Jahre ergeben sich jährliche Amortisationskosten

$$A_{P6} = \frac{I_{HP6}}{d_A} = 27000 \text{ Fr} \quad (8.2)$$

8.3.1.1 Die Kosten des gegenwärtigen PLANETH-Betriebs

Bei $h_{AK} = 1829$ Adressatenkontaktstunden pro Jahr (gegenwärtige Bruttoauslastung) beträgt der Kostenanteil für die Hardwareamortisation

$$k_{AP6} = \frac{A_{P6}}{h_{AK}} = 14,76 \text{ Fr/h} \quad (8.3)$$

Für die Ueberwachung des Lehrbetriebs, das richtige Auswechseln der Magazine, die Betreuung der Adressaten und für kleinere Wartungsarbeiten ist ein Operateur erforderlich, der das Lehrsystem einigermaßen kennt. Für diese Arbeit erwies sich der Einsatz von Hilfsassistenten (Studenten) als günstigste Lösung. Diese Hilfsassistenten beziehen pro Arbeitsstunde $L_{HA} = 12$ Fr/h Lohn. Bei $k = 6$ Konsolen beträgt somit der Betriebskostenanteil

$$k_{BP6} = \frac{L_{HA}}{k} = 2 \text{ Fr/h} \quad (8.4)$$

Die Kosten für die Lehrprogrammherstellung variieren sehr stark (vergl. Kap. 6.4). Je nach Komplexität des Lehrprogramms und Qualifikation des Autors betragen die Kosten für die Herstellung eines einstündigen PLANETH-Lehrprogramms in 6 Kopien ca. 1000 - 8000 Fr. Ein solches Lehrprogramm kann mindestens etwa 10 Jahre benützt werden (vergl. Kap. 1). Bei 200 Durchläufen pro Jahr, die sich beim Einsatz eines Programms in einer grossen propädeutischen Hochschulveranstaltung leicht erreichen lassen, ergibt sich somit eine totale Benützungsdauer d_{BL} von mindestens 2000 Stunden.

Unter der Annahme, dass die mittleren Lehrprogrammherstellungskosten L etwa 2500 Fr betragen, ergibt sich für den Lehrprogramm-Kostenanteil

$$k_L = \frac{L}{d_{BL}} = 1,25 \text{ Fr/h} \quad (8.5)$$

Es muss betont werden, dass diese Kostenangabe für k_L nicht auf hypothetischen Annahmen über teilweise phantastisch anmutende Benützerzahlen beruht, sondern auf Werten, die bei einigen der gegenwärtig eingesetzten PLANETH-Lehrprogramme tatsächlich erreicht werden.

Wenn ein Lehrprogramm an einer Hochschule nicht nur in einem bestimmten Kurs, sondern in mehreren ähnlichen Kursen für verschiedene Fachrichtungen verwendet werden kann (z.B. Programme über Grundlagen der Mathematik oder Physik an einer technischen Hochschule), so lassen sich leicht bis zu 1000 Programmdurchläufe pro Jahr erreichen, wodurch k_L auf etwa 0,25 fr/h sinkt. Aehnliche oder sogar noch kleinere Werte für k_L ergäben sich natürlich, wenn das gleiche Lehrprogramm auf mehreren PLANETH-Systemen an verschiedenen Hochschulen benützt werden könnte.

Bei den Kosten des heutigen PLANETH-Betriebs dominieren infolge der schlechten Gesamtauslastung noch eindeutig die Systemkosten, d.h. die Kosten für die Amortisation und den Betrieb des CUU-Systems :

$$k_{S_{P6}} = k_{A_{P6}} + k_{B_{P6}} = 16,76 \text{ fr/h} \quad (8.6)$$

Die totalen Kosten pro CUU-Stunde belaufen sich beim heutigen nur teilweise ausgelasteten PLANETH-System mit 6 Konsolen somit auf

$$k_{CUU_{P6}} = k_{S_{P6}} + k_L = 18 \text{ fr/h} \quad (8.7)$$

Diese Kosten sind natürlich noch wesentlich höher als beim konventionellen Unterricht in Vorlesungen oder Uebungsgruppen. Wenn man jedoch berücksichtigt, dass das individuelle Durcharbeiten eines PLANETH-Programms eigentlich eine Art Privatstunde beim Programmautor mit entsprechend grossem Lehrwirkungsgrad darstellt, erscheinen diese Kosten noch durchaus vertretbar, denn sie liegen deutlich unter den Kosten einer solchen Privatstunde.

8.3.1.2 Die Betriebskosten des 6-Konsolen-Systems bei Volllauslastung

Bei Volllauslastung des gegenwärtig vorhandenen 6-Konsolen-Systems während 2500 Stunden pro Jahr könnten jährlich $h_{AK} = 15000$ Adressatenkontaktstunden absolviert werden, so dass der Kostenanteil für die Hardwareamortisation auf folgenden Wert sinken würde :

$$k_{A_{P6v}} = \frac{A_{P6}}{h_{AK}} = 1,8 \text{ Fr/h} \quad (8.8)$$

k_B und k_L blieben unverändert, so dass sich für das heutige 6-Konsolen-PLANETH-System folgende Kosten pro Betriebsstunde ergeben würden :

$$k_{CUU_{P6v}} = k_{A_{P6v}} + k_{B_{P6}} + k_L \approx 5 \text{ Fr/h} \quad (8.9)$$

Diese Kosten sind durchaus vergleichbar mit denjenigen für eine konventionelle Übungsstunde :

Wenn ein Assistent, der die Hochschule inkl. Sozialleistungen und Arbeitsplatz 30 Fr/h kostet und der sich für jede Übungsstunde 2 Stunden vorbereiten muss, eine Gruppe von 15 Studenten betreut, so entstehen pro Student Kosten von 6 Fr/h .

8.3.2 Die Kosten eines voll ausgelasteten 32-Konsolen-PLANETH-Systems

Bei Serienproduktion der PLANETH-Konsolen könnten die Herstellungskosten noch beträchtlich gesenkt werden. Ein Preis von 10000 Fr pro Konsole sollte dabei erreichbar sein. Der zentrale Kleincomputer mit der zugehörigen Standardperipherie kommt heute nur noch auf etwa 30000 Fr zu stehen. Die Investitionskosten für ein komplettes 32-Konsolen-PLANETH-System berechnen sich somit wie folgt :

Computer (inkl. Standardperipherie)	30000 Fr	
Vorschaltgerät inkl. Party-Line	10000 Fr	
32 Konsolen zu 10000 Fr	<u>320000 Fr</u>	
Totale Investitionskosten $I_{HP32} =$	360000 Fr	(8.10)

Bei $d_A = 5$ Jahren Amortisationsdauer betragen die jährlichen Amortisationskosten somit

$$A_{P32} = \frac{I_{HP32}}{d_A} = 72000 \text{ Fr} \quad (8.11)$$

Bei 2500 Betriebsstunden pro Jahr können jährlich $h_{AK} = 80000$ Adressatenkontaktstunden absolviert werden, d.h. der Kostenanteil für die Hardwareamortisation sinkt auf

$$k_{A_{P32v}} = \frac{A_{P32}}{h_{AK}} = 0,9 \text{ Fr/h} \quad (8.12)$$

Dank der grösseren Konsolenzahl $k = 32$ sinkt auch der Betriebskostenanteil (vergl. Formel (8.4)):

$$k_{B_{P32}} = \frac{L_{HA}}{k} \approx 0,4 \text{ Fr/h} \quad (8.13)$$

Die Systemkosten für Amortisation und Betrieb eines voll ausgelasteten 32-Konsolen-PLANETH-Systems betragen somit nur noch

$$k_{S_{P32v}} = k_{A_{P32v}} + k_{B_{P32}} = 1,3 \text{ Fr/h} \quad (8.14)$$

Dieser Wert ist durchaus vergleichbar mit den Kosten anderer CUU-Systeme (siehe Kap. 9).

Mit $k_L = 1,25 \text{ Fr/h}$ (200 Durchläufe pro Jahr) ergeben sich somit für ein voll ausgelastetes 32-Konsolen-System :

$$k_{CUU_{P32v}} = k_{S_{P32v}} + k_L = 2,55 \text{ Fr/h} \quad (8.15)$$

Mit $k_L = 0,25 \text{ Fr/h}$ bei 1000 Durchläufen pro Jahr sinkt k_{CUU} sogar auf

$$k'_{CUU_{P32v}} = 1,55 \text{ Fr/h} \quad (8.16)$$

Diese Kostenansätze für eine Stunde CUU liegen weit unter den Kosten des konventionellen Uebungsbetriebs (siehe Kap. 8.3.1.2).

Ein voll ausgelastetes 32-Konsolen-PLANETH-System ist also im Uebungseinsatz sicher wirtschaftlich. Seine Kosten liegen sogar so tief, dass sie durchaus vergleichbar mit der Kosten des konventionellen Unterrichts an Mittelschulen oder mit den Kosten von Hochschulvorlesungen sind. Dank der relativ niedrigen Anfangsinvestition von 360000 Fr dürfte ein solches CUU-System für Hochschulen und grössere Mittelschulen noch erschwinglich sein.

9. VERGLEICH ZWISCHEN PLANETH UND EINIGEN ANDERN CUU-SYSTEMEN

9.1 Vergleich mit andern kleincomputergesteuerten CUU-Systemen

Von der Konzeption und der technischen Realisierung her lässt sich PLANETH am ehesten mit den CUU-Systemen EDUCATOR [15] und SWISSAIR [30] vergleichen. In Tab. 9.1 sind die wesentlichen Merkmale dieser drei Systeme zusammengestellt. Alle drei Systeme verwenden Computer mit ähnlicher Kernspeichergrösse. Bei EDUCATOR ist der Aufwand beim Computer und bei den Lehrkonsolen etwas geringer als bei den beiden andern Systemen: Der Kernspeicher des Computers ist etwas kleiner und der zweite Projektor fehlt. Dadurch entfällt bei EDUCATOR natürlich die Möglichkeit der Doppelprojektion. Alle drei Systeme verwenden auch ähnliche oder gleiche Projektoren und Diamagazine. Ebenso sind alle drei Systeme für vertonte Programme mit maximal 127 verschiedenen Tonsequenzen geeignet. Die verwendeten Tonträger sind jedoch verschieden: PLANETH benützt die heute weit verbreiteten Kompaktkassetten, EDUCATOR etwas mühsamer auszuwechselnde Spulentonbänder, während im SWISSAIR-System Spezialkassetten mit sehr breitem Band und 127 parallelen Spuren eingesetzt werden. Für die Anzeige häufig vorkommender Meldungen besitzt PLANETH 6 und EDUCATOR 3 spezielle Anzeigelampen, während beim SWISSAIR-System einige solche Meldungslampen in die Tasten integriert sind. Bei PLANETH besteht zusätzlich noch die Möglichkeit, computergenerierte Zahlen auf den Konsolen anzuzeigen.

Die Ausgabemöglichkeiten der Lehrkonsolen sind also beim SWISSAIR-System ähnlich wie bei PLANETH, wenn man von der numerischen Anzeigemöglichkeit absieht. Bei EDUCATOR sind diese Möglichkeiten etwas weniger gut ausgebaut, jedoch noch durchaus vergleichbar mit den beiden andern Systemen.

Beträchtliche Unterschiede zwischen den drei Systemen bestehen jedoch bei den Möglichkeiten zur Verarbeitung der Adressatenreaktionen. Während bei EDUCATOR und beim SWISSAIR-System nur Fragen mit höchstens 4 Auswahlantworten zulässig sind, verarbeitet PLANETH solche mit 5 Auswahlantworten. Als einziges der drei Systeme lässt PLANETH auch Fragen mit numerischen Freiwahlantworten im Bereich von $\pm 10^{-38}$ bis $\pm 10^{+38}$ und 0 zu, wobei nicht nur ein einfacher Vergleich richtig-falsch erfolgt, sondern eine eingehende Analyse des eingegebenen Zahlenwertes (siehe Kap. 6.6). Auch bei der Bewertung der eingegebenen Antworten ist PLANETH viel flexibler, da es eine sehr differenzierte Bewertung in maximal 16 Stufen zulässt, während die beiden andern Systemen nur eine Unterscheidung zwischen richtigen und falschen Antworten kennen. PLANETH ist somit bei der Verarbeitung von Adressatenreaktionen den Systemen EDUCATOR und SWISSAIR eindeutig überlegen.

Auch bei der Software, also beim Betriebssystemprogramm und den zugehörigen Hilfsprogrammen, bestehen beträchtliche Unterschiede. Das SWISSAIR-System ist bezüglich Software am wenigsten gut ausgerüstet. Es erlaubt nur markoffsche (intrinsische) Verzweigungen (siehe Kap. 1.1) und druckt auf speziellen Wunsch nur gerade ein laufendes Protokoll der an einer einzigen Lehrkonsole erfolgenden Adressatenreaktionen auf dem Fernschreiber aus. Eine statistische Analyse aller Lehrprogrammdurchläufe ist somit nicht möglich. Immerhin verfügt auch dieses System über einen einfachen Compiler, d.h. ein Uebersetzungsprogramm für die Erzeugung des Lehrprogrammcodes.

EDUCATOR besitzt eine bessere Software. Als Besonderheit ist bei diesem System der Lehrprogrammcode, also die Information über Struktur und Ablauf des Lehrprogramms, jeweils nicht im Kernspeicher gespeichert, sondern befindet sich auf der 2. Spur des Tonbandes. Auf diese Weise kann Computerspeicherplatz eingespart werden, dafür ist jedoch zusätzliche Elektronik notwendig, die Aenderungen an Betriebssystem und Lehrprogrammen unnötig kompliziert macht. Auch ist bei EDUCATOR-Lehrprogrammen deshalb immer ein Tonband notwendig, selbst bei unvertonten Programmen, denn die Programmsteuerinformationen müssen im Betrieb vom Tonband gelesen werden können. Die Herstellung von Lehrprogrammen ist bei EDUCATOR deshalb aufwendiger als bei PLANETH, wo Tonkassetten nur für effektiv vertonte Programme benötigt werden und sich der Aufwand somit nach der Komplexität des Lehrprogramms richtet (vergl. Kap. 6.4). Der Computer interpretiert den auf dem Tonband gespeicherten Programmcode und steuert den Lehrprozess entsprechend. Etwa die Hälfte des Kernspeichers wird zur Speicherung der während des Lehrprozesses anfallenden Lernverhaltensdaten verwendet. Auf Grund dieser Daten sind bei EDUCATOR verschiedene Arten von nichtmarkoffschen Programmverzweigungen möglich. Am Schluss jedes Programmdurchlaufs können die entsprechenden Verhaltensdaten als Schlussprotokoll auf dem Fernschreiber ausgedruckt werden. Dieses Protokoll kann beim verwendeten Fernschreiber auf Wunsch gleichzeitig auf einen Lochstreifen gestanzt werden. Es ist unklar, ob ein Statistikauswerteprogramm für solche Streifen existiert und welche Möglichkeiten es allenfalls bietet. In [15] wird jedenfalls kein solches Programm erwähnt. Auch EDUCATOR verfügt über einen einfachen Compiler für die Erzeugung des Lehrprogrammcodes.

	<u>PLANETH</u>	<u>EDUCATOR</u>	<u>SWISSAIR</u>
<u>Vergleich</u> :			
<u>Computer</u> :	Hersteller/Typ Speichergrösse	Hewlett-Packard 2116 Wagner Comp. WAC 12 8K 12-Bit-Worte	Gen. Automation SPC 12 16K 8-Bit-Worte
<u>Lehrkonsolen</u> :	Maximale Anzahl Projektoren Max. Anzahl Dias Tonträger Max. Tonsequenzzahl Anzahl Tasten total Anzeigelampen Auswahlantworttasten Freiwahlantworten Anzeige von Zahlen	32 2 Kodak S-RA 160 Kompakt-Kassette 127 19 6 5 ja, numerische ja	16 1 Kodak S-RA 80 Tonband (1.Spur) 127 10 3 4 nein nein
<u>Betriebssystem</u> :	Lehrprogrammstruktur Lehrprogrammanzahl Antwortbewertung Statistikdaten Schlussprotokoll Nichtmarkoff-Verzw. Zufallsverzweigungen Unterprogramme	im Kernspeicher ≤16 (je nach Länge) 16 Stufen (0-15P) Ausg. auf Lochstreifen ja ja ja ja	im Kernspeicher 13 nur richtig/falsch nur Protokoll für 1Kons. nein ? nein nein nein
<u>Hilfsprogramme</u> :	Compiler Statistik-Auswertung	ja, komfortabel ja, komfortabel	ja nein

Tab. 9.1 : Vergleich der Lehrsysteme PLANETH-EDUCATOR-SWISSAIR

Das PLANETH-System besitzt eine sehr gut ausgebaute Software. Wie beim SWISSAIR-System ist der Lehrprogrammcode im Kernspeicher gespeichert. Dabei ist jedoch nicht für jedes Lehrprogramm ein Bereich konstanter Länge reserviert, was nur eine schlechte Ausnützung des kostbaren Speichers erlauben würde, sondern es sind alle Lehrprogramme unmittelbar hintereinander gespeichert und belegen jeweils nur gerade den ihrer Länge entsprechenden Speicherplatz. Während des Lehrbetriebs können einzelne Programme ohne Störung desselben ausgewechselt werden, wobei die verbleibenden Programme jeweils wieder lückenlos zusammengeschoben werden. Auf diese Weise wird bei PLANETH der vorhandene Speicherplatz immer optimal ausgenützt.

Auch bei PLANETH sind nichtmarkoffsche Verzweigungen möglich (siehe Kap. 2.2). Zudem erlaubt PLANETH als einziges der drei Systeme Zufallsverzweigungen und Unterprogramme. Während des Lehrbetriebs werden bei PLANETH die für eine spätere statistische Auswertung relevanten Daten aller Konsolen laufend auf Lochstreifen ausgestanzt. Ein umfangreiches Statistikauswerteprogramm erlaubt die eingehende Analyse dieser Statistikdaten nach vielen verschiedenen Gesichtspunkten (siehe Kap. 7). Der Compiler zum PLANETH-System ist sehr komfortabel aufgebaut (siehe Kap. 6.5). Die Lehrprogrammbeefehle bestehen bei PLANETH nicht aus schwer lesbaren Buchstaben- und Zahlengruppen, die zuerst mühsam auf Codierblättern zusammengestellt werden müssen, sondern sie sind in einer gut verständlichen und leicht erlernbaren Autorensprache (SESAM, siehe Kap. 6.6) geschrieben, die manche Aehnlichkeiten mit andern Computersprachen besitzt.

Die oben durchgeführten Vergleiche zeigen, dass PLANETH eindeutig mehr Möglichkeiten bietet und die einem Kleincomputer-CUU-System gesteckten Grenzen besser ausnützt als die beiden Vergleichssysteme. Es ist jedoch klar, dass solche Kleincomputerlehrsysteme nicht für alle Lehrstoffe gleich gut geeignet sind. Besonders vorteilhaft können sie für die grundlegende Fachausbildung in naturwissenschaftlichen und technischen Fächern eingesetzt werden. Solche Systeme eignen sich infolge der beschränkten Eingabemöglichkeiten weniger für den Sprachunterricht, für den Unterricht im Programmieren und für Lehrstoffe, die nichtnumerische Freiwahlantworten erfordern. Derartige Kleincomputerlehrsysteme können den Lehrer auch niemals vollständig ersetzen, aber sie können ihn bei der Vermittlung und beim Einüben grundlegender Kenntnisse wesentlich entlasten und den Unterricht wirksamer und interessanter gestalten.

9.2 Beispiele einiger grosser CUU-Systeme

Die in Kap. 9.1 beschriebenen audiovisuellen Kleincomputer-CUU-Systeme eignen sich nicht für alle Lehrstoffe gleich gut. In diesem Kapitel sollen die beiden bekanntesten Gross-CUU-Systeme PLATO IV und TICCIT kurz vorgestellt werden. Diese Lehrsysteme erfordern einen wesentlich höheren technischen und finanziellen Aufwand als Kleincomputersysteme, sie sind dafür aber dank der vielseitigen Ein- und Ausgabemöglichkeiten für sehr viele Lehrstoffe geeignet.

9.2.1 Das PLATO IV - System

Das PLATO IV-System der University of Illinois (USA) wird von einem sehr grossen und leistungsfähigen Computer der Serie CDC 6000 gesteuert. Sein Kernspeicher umfasst insgesamt etwas mehr als 2 Millionen Worte zu 60 Bits. An diesem Computer sollten nach den ursprünglichen Plänen maximal 4000 Lehrkonsolen angeschlossen werden können [34]. Im praktischen Versuchsbetrieb zeigte es sich jedoch, dass die Systemplanung doch etwas zu optimistisch war und dass dieser Computer effektiv nur etwa 1000 Terminals bedienen kann [35, 57]. Die Statusinformationen der Terminals und die gerade benützten Lektionen sind jeweils im Kernspeicher vorhanden, so dass die Zugriffszeit während des eigentlichen Lehrbetriebs sehr klein ist. Daneben stehen zusätzliche Plattenspeicher für die Speicherung von mehreren tausend gerade nicht benützten Lehrprogrammen zur Verfügung.

Die Datenübertragung zwischen Computer und Konsolen kann über normale Telefonleitungen mit einer Geschwindigkeit von 1200 Bit/s erfolgen. Die Konsolen müssen sich also nicht in unmittelbarer Nähe des Computers befinden. Mit zunehmender Entfernung steigen aber natürlich die Kosten für die Leitungsmiete.

Das PLATO IV-Studenten-Terminal besitzt sehr vielseitige Ein- und Ausgabemöglichkeiten. Der speziell für diesen Zweck entwickelte Plasma-Display-Bildschirm erlaubt die Uebereinanderprojektion von maximal 256 auf Mikrofilm gespeicherten farbigen Bildern mit computergenerierten dynamischen Texten und grafischen Darstellungen (maximal 512·512 Punkte). Die vom Computer kommende Bildinformation wird dabei direkt im Plasma-Display gespeichert, es ist somit kein spezieller Speicher notwendig. Zusätzlich können pro Lektion bis zu 4096 verschiedene Tonsequenzen mit einer Gesamtlänge von 21 Minuten abgespielt werden [34].

Bei PLATO IV wurde speziell auf eine möglichst ökonomische Ausnützung des Kernspeichers und der Datenübertragungsleitungen geachtet. Deshalb sind die statischen farbigen Bilder und die Tonsequenzen, für deren digitale Speicherung sehr

viel Speicherplatz erforderlich wäre, in analoger Form dezentral in den Konsolen gespeichert, während der Lehrprogrammcode und die für die Erzeugung der dynamischen Grafiken und Texte nötigen Daten im Kernspeicher des Computers untergebracht sind.

Die Eingabe der Studentenantworten erfolgt meist über eine alphanumerische Tastatur, kann jedoch auch durch Berühren einzelner Stellen des Bildschirms erfolgen (Eingabe mit dem sogenannten Touch-Panel mit 256 Feldern). Für die Verarbeitung der Antworten stehen viele leistungsfähige Unterprogramme zur Verfügung. Bei aus Text bestehenden Antworten erlauben diese neben dem Exaktvergleich und dem Suchen von Schlüsselwörtern auch die Erkennung kleinerer orthografischer Fehler. Bei der Textverarbeitung müssen jedoch neben allen richtigen auch die typischen falschen Antworten vom Programmautor im voraus angegeben werden, damit das System die Antwort richtig beurteilen kann.

Die Herstellung der Lehrprogramme ist von jeder Konsole aus während des normalen Lehrbetriebs möglich. Dem Autor steht zu diesem Zweck eine relativ leicht zu erlernende Autorensprache (TUTOR) mit zugehörigem Uebersetzungsprogramm (Compiler) zur Verfügung. Natürlich können vom Terminal aus nur der Lehrprogrammcode und die Daten für die Ausgabe der computergenerierten Texte und Grafiken erzeugt werden. Die Herstellung farbiger Bilder auf Mikrofilm und eventueller Tonaufnahmen muss separat erfolgen.

Bei PLATO IV ist die Herstellung von Lehrprogrammen also noch mit relativ einfachen Mitteln möglich. Der Aufwand wächst natürlich mit steigender Komplexität des Programms. Ein einzelner Professor oder Lehrer kann durchaus noch selbst ein Programm für eine von ihm durchgeführte Lehrveranstaltung schreiben. PLATO IV stellt somit einen Versuch dar, die Produktivität des einzelnen Lehrers und den Wirkungsgrad des Unterrichts zu erhöhen, ohne den Lehrer jedoch vollständig zu ersetzen. Das PLATO IV-System ist sehr vielseitig, es können damit bei entsprechender Programmierung alle in Kap. 1.3.3 erwähnten Lehrstrategien realisiert werden.

PLATO IV ist schon einige Jahre im Gebrauch und wurde bisher vorwiegend für den Hochschulunterricht verwendet. Die reinen Hardware-Investitionskosten für ein PLATO IV-System mit 1000 Konsolen belaufen sich auf etwa 10 Millionen Dollar. Dazu kommen die Unterhaltskosten, die Kosten für die Miete der Datenübertragungsleitungen und die Kosten für die Erstellung der Lehrprogramme. Eine ausführlichere Analyse der Kosten beim PLATO IV-System befindet sich in Kap. 9.3 .

9.2.2 Das Lehrsystem TICCIT

Im Gegensatz zu PLATO IV wird das TICCIT-System von zwei wesentlich kleineren Computern gesteuert [32, 58]. Zur Organisation des Datenaustauschs mit den maximal 128 Konsolen dient ein NOVA 800 Computer mit einer Speichergrösse von 32K 16-Bit-Worten.

Die TICCIT-Konsolen bestehen aus Farbfernsehmonitoren mit Kopfhörer und alphanumerischer Tastatur. Diese Terminals sind über Videokabel mit zusätzlichen Adern für die Uebertragung von Ton und Tasteninformation mit dem Computer verbunden. Zur Speicherung der vom Computer kommenden Bildinformation in der Konsole wird ein integrierter Halbleiterspeicher verwendet.

Bei TICCIT ist ausser dem Lehrprogrammcode auch fast der gesamte Lehrstoff (statische Bilder und Ton) auf insgesamt 5 grossen Plattenspeichern untergebracht. Der Zugriff zu diesen Lehrstoffspeichern, die Aufbereitung der Lehrinformation und die Verarbeitung der Adressatenantworten erfolgt mit Hilfe eines zweiten NOVA 800 Computers mit 48K Speichergrösse. Die beiden Computer verfügen zudem über sehr viel Standardperipherie (Bildschirm-Datenstation, Liniendrucker, Kartenleser und Magnetbandstation) sowie einige spezielle Geräte für die analog-digital- und die digital-analog-Umwandlung der Bild- und Tonsignale. Im Gegensatz zu PLATO IV erlaubt TICCIT auch die Wiedergabe kurzer Sequenzen von bewegten farbigen Bildern, die auf Videobändern in analoger Form gespeichert sind.

Wie bei PLATO erfolgt die Ausgabe der Lehrinformation an den Adressaten bei TICCIT vorwiegend visuell. Maximal sind auf dem Bildschirm 17·43 alphanumerische Zeichen oder Grafiken mit 204·430 Punkten darstellbar. Ton kann nur ergänzend und während relativ kurzer Zeit verwendet werden, denn für die digitale Speicherung von Ton wird enorm viel Speicherplatz benötigt. Auch bewegte farbige Bilder können nur beschränkt eingesetzt werden, da lange nicht für jede Konsole ein Video-Abspielgerät zur Verfügung steht.

Die Herstellung von TICCIT-Lehrprogrammen ist technisch wesentlich komplizierter als bei PLATO IV. Deshalb werden TICCIT-Lehrprogramme nicht mehr durch die direkt am Unterricht beteiligten Professoren oder Lehrer produziert, sondern durch spezialisierte Teams aus Programmierern, Grafikern, Psychologen und Fachleuten des entsprechenden Lehrgebietes. Auf diese Weise entstehen wohl meist Lehrprogramme von hoher Qualität, aber zu Kosten, die um ein Vielfaches höher sind als bei PLATO IV. TICCIT-Lehrprogramme müssen somit von noch viel grösseren Adressatenzahlen durchgearbeitet werden als PLATO IV-Programme, damit der Programmkostenanteil für den praktischen Betrieb klein genug ist. TICCIT wurde ursprünglich speziell für den Einsatz in Mittelschulen entwickelt. Auf dieser

Stufe sind die erforderlichen sehr grossen potentiellen Adressatenzahlen vorhanden und die Anzahl der Fächer ist noch relativ beschränkt, so dass der Einsatz von TICCIT-Programmen durchaus wirtschaftlich sein kann. Nach [32] sollen die Kosten für ein komplettes TICCIT-System mit 128 Konsolen etwa 450000 US-Dollar (Preisstand 1972) betragen, d.h. die erforderlichen Hardware-Investitionskosten liegen in einem Bereich, der für solche Mittelschulen gerade noch tragbar sein dürfte.

Das TICCIT-Lehrsystem soll den konventionellen Unterricht in manchen Fächern weitgehend ersetzen und nicht nur ergänzen können [53]. Der Lernende kann bei TICCIT in einem gewissen Umfang die Zusammensetzung des Lehrstoffs selbst beeinflussen, es ist also ein gewisses Mass von Lernersteuerung möglich. In der Übungsphase kann der Lernende beispielsweise durch Drücken entsprechender Tasten leichtere oder schwerere Übungsaufgaben anfordern, bei einem ihm schwierig erscheinenden Problem zusätzliche Hilfe verlangen oder den Rat des Systems über das zweckmässige weitere Vorgehen im Lehrprogramm erbitten. Der Lehrer hat somit beim Einsatz von TICCIT-Programmen oft nur noch Ueberwachungs- und Beratungsaufgaben. Es ist klar, dass ein derartiges CUU-Konzept auf beträchtlichen Widerstand seitens der Lehrerschaft stossen dürfte.

Nach den vorliegenden Informationen stehen in den USA heute einige TICCIT-Systeme an Mittel- und Hochschulen im praktischen Einsatz. Diese Systeme haben aber alle wesentlich weniger als die ursprünglich vorgesehenen 128 Konsolen. An TICCIT-Lehrmaterial wurden bisher unter anderem je 2 einsemestrige Kurse in Englisch und Mathematik entwickelt. Die Produktionskosten dieser Kurse sollen sich auf insgesamt 1½ Millionen US-Dollar belaufen haben.

Zusammenfassend kann man feststellen, dass TICCIT ähnliche didaktische Möglichkeiten bietet wie PLATO IV. Die erforderlichen Hardware-Investitionskosten sind bei TICCIT geringer. Dafür sind die Kosten für die Lehrprogrammherstellung bei TICCIT beträchtlich höher als bei PLATO IV.

9.3 Die Kosten bei PLATO IV im Vergleich zu den Kosten bei PLANETH

PLATO IV bietet natürlich didaktisch wesentlich mehr Möglichkeiten als PLANETH. In diesem Kapitel sollen nur die Kosten der beiden CUU-Systeme miteinander verglichen werden. Es wird sich dabei zeigen, dass die in Kap. 8.3.2 berechneten Kosten pro CUU-Stunde bei einem voll ausgelasteten PLANETH-System mit 32 Konsolen niedriger sein dürften als

beim gegenwärtig im Betrieb stehenden PLATO IV-System der University of Illinois (USA), das in der Literatur allgemein als sehr wirtschaftlich gilt. Für diese Kostenschätzung wird wieder Formel (1.1) verwendet. Dabei wird berücksichtigt, dass nach [35] das heutige PLATO IV-System effektiv nur etwa 1000 Konsolen richtig bedienen kann. Die folgenden Kostenangaben sind zum heutigen Kurs 1 US-Dollar \approx 2 fr in Schweizer Franken umgerechnet, damit eine direkte Vergleichsmöglichkeit mit den in Kap. 8.3 angegebenen Werten möglich ist.

Die Hardware-Investitionskosten I_H setzen sich bei PLATO IV wie folgt zusammen [35, 54]:

Computer zu 4,5 Millionen Dollar	9000000 fr	
1000 Konsolen zu 5500 Dollar	<u>11000000 fr</u>	
Hardware-Investitionskosten $I_{H_{PLATO}}$	=	20000000 fr (9.1)

Bei einer fünfjährigen Amortisationsdauer d_A ergeben sich jährliche Hardware-Amortisationskosten

$$A_{PLATO} = 4000000 \text{ fr} \quad (9.2)$$

Bei einer jährlichen Betriebsdauer von 2500 Stunden und 1000 Konsolen ergeben sich bei einem voll ausgelasteten PLATO IV-System pro Jahr $h_{AK} = 2500000$ Adresatenkontaktstunden. Somit beträgt der Kostenanteil für die Hardwareamortisation bei PLATO IV :

$$k_{A_{PLATO}} = 1,6 \text{ fr/h} \quad (9.3)$$

Die jährlichen Betriebskosten setzen sich aus den Betriebs- und Unterhaltskosten des PLATO IV-Rechenzentrums, den Kosten für die Miete der Datenübertragungsleitungen und den Unterhaltskosten für die 1000 PLATO IV-Konsolen zusammen. Für die jährlichen Betriebs- und Unterhaltskosten (central management services) sind in [54] 240000 Dollar angegeben. Da es sich bei dieser Angabe vorwiegend um Personalkosten mit Preisstand 1970 handelt, ist eine gewisse inflationsbedingte Aufwertung dieses Betrags auf etwa 900000 fr (Preisstand 1978) sicher gerechtfertigt. Die Kosten für die Datenübertragung zwischen Computer und Konsolen kommen nach [35] wesentlich höher zu stehen als in [54] angenommen, wobei genaue Angaben allerdings fehlen. Es ist wahrscheinlich vertretbar, für alle Konsolen mit dem in [54] angegebenen Höchstwert für die Leitungskosten zu rechnen, was insgesamt jährliche Kosten für Leitungsmiete von 100000 fr ergibt.

Damit ergeben sich bei PLATO IV folgende jährliche Betriebskosten :

$$B_{PLATO} = 1000000 \text{ fr} \quad (9.4)$$

Daraus folgt bei $h_{AK} = 2500000$ Adressatenkontaktstunden pro Jahr ein Betriebskostenanteil

$$k_{B_{PLATO}} = 0,4 \text{ Fr/h} \quad (9.5)$$

Die Systemkosten für Amortisation und Betrieb betragen bei einem voll ausgelasteten PLATO IV-System mit 1000 Konsolen somit

$$k_{S_{PLATO}} = k_{A_{PLATO}} + k_{B_{PLATO}} = 2 \text{ Fr/h} \quad (9.6)$$

Zum Vergleich: Bei einem voll ausgelasteten PLANETH-System mit 32 Konsolen betragen die Systemkosten nach (8.14) nur 1,3 Fr/h.

Die mittleren Kosten für die Herstellung einer einstündigen PLATO IV-Lektion werden in [54] mit 600 - 800 Dollar (Preisstand 1970) und in [35] mit etwa 1000 Dollar (Preisstand Frühling 1975) angegeben. Da es sich hier wiederum vorwiegend um Personalkosten handelt, ist eine gewisse Aufwertung entsprechend der seither eingetretenen Teuerung angebracht. Die mittleren Kosten (Preisstand 1978) für die Herstellung einer einstündigen PLATO IV-Lektion werden damit etwa gleich gross wie bei PLANETH, d.h.

$$L_{PLATO} = 2500 \text{ Fr} \quad (\text{für eine einstündige Lektion}) \quad (9.7)$$

In der Literatur über PLATO IV wird nun für die Berechnung des Lehrprogrammkostenanteils k_L oft von teilweise phantastisch anmutenden Benutzerzahlen ausgegangen. Um einen k_L -Wert von 3 Cents = 6 Rp. pro Stunde zu erhalten, müsste eine PLATO IV-Lektion von über 40000 Benutzern absolviert werden. Solche Zahlen wären beim Einsatz von PLATO IV-Lektionen in nicht allzu weit entfernten Elementarschulen schon bei einem einzigen PLATO IV-System möglicherweise erreichbar. Wegen der relativ hohen Systemkosten gemäss Formel (9.6) kommt aber eine wirtschaftlich vertretbare Verwendung von PLATO IV-Unterrichtsmaterial auf dieser Stufe zu den heutigen Kosten wohl kaum in Frage.

Auf Hochschulstufe andererseits sind die bei einem einzelnen PLATO IV-System vorhandenen potentiellen Benutzerzahlen wesentlich geringer. In einigen wenigen sehr stark belegten Fächern kann an einer sehr grossen Universität vielleicht mit jährlich 2000 - 3000 Benutzern gerechnet werden. Diese paar Fächer reichen jedoch niemals aus, um ein derart grosses CUU-System wie PLATO IV auszulasten. In vielen anderen Fachgebieten werden die jährlichen Benutzerzahlen nur etwa zwischen 100 und 1000 liegen. Somit dürfte das heute existierende

PLATO IV-System in Wirklichkeit ähnliche Ansätze für den Lehrprogrammkostenanteil haben wie das PLANETH-System :

Bei einer Nutzungsdauer von 10 Jahren für eine PLATO IV-Lektion und jährlich 200 Benützern wird $k_L = 1,25 \text{ Fr/h}$, bei jährlich 1000 Benützern dagegen $k_L = 0,25 \text{ Fr/h}$.

Falls die gleiche PLATO IV-Lektion an mehreren Hochschulen verwendet werden könnte, ergäben sich natürlich noch kleinere Werte für k_L . Dazu wäre jedoch eine gewisse minimale Koordination zwischen den betreffenden Hochschulen notwendig, die infolge des foederalistisch aufgebauten Bildungssystems in vielen westlichen Staaten nur mit Mühe realisierbar sein dürfte.

Für das heute im Betrieb stehende PLATO IV-System mit 1000 Konsolen ergeben sich bei Vollauslastung mit $k_L = 1,25 \text{ Fr/h}$ (200 Benutzer pro Jahr) somit folgende Gesamtkosten pro Stunde CUU :

$$k_{\text{CUU PLATO}} = k_{\text{S PLATO}} + k_L = 3,25 \text{ Fr/h} \quad (9.8)$$

Mit $k_L = 0,25 \text{ Fr/h}$ bei 1000 Benützern pro Jahr sinkt k_{CUU} auf

$$k'_{\text{CUU PLATO}} = 2,25 \text{ Fr/h} \quad (9.9)$$

Die oben durchgeführten Berechnungen können natürlich nicht den Anspruch auf sehr grosse Genauigkeit erheben, denn sie beruhen auf verschiedenen Annahmen und zum Teil mehrere Jahre alten Literaturangaben. Sie ergeben aber doch einen Anhaltspunkt für die Grössenordnung der bei einem voll ausgelasteten PLATO IV-System auftretenden Kosten.

Nach Auskunft von Prof. J. Nievergelt, ETH-Zürich, der das PLATO IV-System der University of Illinois einigermaßen kennt, ist das System heute nur relativ schwach ausgelastet. Von den etwa 1000 Konsolen werden in Spitzenzeiten (ca. 2 - 3 Stunden pro Tag) vielleicht 500 gleichzeitig benützt, in den Randzeiten jedoch noch wesentlich weniger. Die effektiven Systemkosten liegen heute somit noch beträchtlich über dem in (9.6) erhaltenen Wert.

Bei einem voll ausgelasteten PLATO IV-System liegen die Kostenansätze gemäss (9.8) und (9.9) so tief, dass PLATO IV-Lektionen im Hochschulunterricht in vielen Fällen wirtschaftlich eingesetzt werden können. Bei einem PLANETH-System mit 32 Konsolen liegen die Kosten gemäss den Formeln (8.15) und (8.16) noch etwas tiefer. PLATO IV bietet aber mehr Möglichkeiten und erlaubt vielseitigere Lehrstrategien, so dass das Kosten/Nutzen-Verhältnis unter Umständen doch noch etwas günstiger ist als bei PLANETH. Ein wesentlicher Vorteil von PLANETH gegenüber PLATO IV ist jedoch die viel kleinere Anfangsinvestition, die auch kleineren Institutionen mit beschränkten finanziellen Möglichkeiten den Einsatz von CUU aus eigener Kraft erlaubt.

10. SCHLUSSBETRACHTUNGEN

Das Ziel dieser Arbeit war die Entwicklung und die ausführliche praktische Erprobung eines für den naturwissenschaftlichen und technischen Hochschulunterricht geeigneten kleincomputergesteuerten Lehrsystems.

Das realisierte Lehrsystem PLANETH bietet wesentlich mehr Möglichkeiten als vergleichbare Kleincomputerlehrsysteme. Es hat sich in über 3½-jährigem Einsatz bewährt, funktioniert zuverlässig und findet bei den Benutzern grossen Anklang.

PLANETH-Lehrprogramme stellen eine zusätzliche, attraktive Unterrichtsform dar, welche die klassischen Unterrichtsmethoden ergänzen und bereichern, keinesfalls jedoch völlig ersetzen kann. Der einzelne Student setzt sich im Dialog mit dem Computer viel intensiver mit dem Lehrstoff auseinander als beim konventionellen Unterricht.

Diese Lehrprogramme sind zu beliebigen Zeiten verfügbar und können auf Wunsch mehrmals wiederholt werden. Durch den Einsatz des Lehrsystems PLANETH können Professoren und Assistenten vor allem bei der Vermittlung von grundlegenden Kenntnissen entlastet werden. Dadurch erhalten sie die Möglichkeit, ihre Zeit vermehrt für die Entwicklung der konstruktiven und synthetischen Fähigkeiten ihrer Schüler einzusetzen, eine Aufgabe, die auch das beste CUU-System nicht erfüllen kann.

Im Rahmen dieser Arbeit wurden keine Untersuchungen über den Lehrwirkungsgrad von PLANETH-Lehrprogrammen durchgeführt. Die Benützung des Lehrsystems PLANETH war für alle beteiligten Studenten freiwillig und konnte auch anonym erfolgen. Eine sichere Unterscheidung zwischen PLANETH-Benützern und -Nichtbenützern war deshalb nicht möglich. Mit entsprechendem Aufwand, d.h. einer strengen Einteilung der Studenten eines Jahreskurses in je eine Gruppe mit und ohne PLANETH-Benützung und der Durchführung von Vor- und Nachtests könnte eine solche Untersuchung sicher interessante Ergebnisse liefern. Infolge der grossen Beliebtheit von PLANETH dürfte es jedoch recht schwierig sein, ohne Zwangsmassnahmen genügend auf die Systembenützung verzichtende Studenten für eine derartige Untersuchung zu erhalten: Im Wintersemester 1977/78 absolvierte von gegen 200 Studenten des Kurses "Technische Elektrizitätslehre III" im Durchschnitt jeder Student 8 von 10 der über dieses Fach vorhandenen Lehrprogramme, d.h. die freiwillige Beteiligung erreichte etwa 80% des theoretisch möglichen Maximums.

Das vorhandene Lehrsystem PLANETH könnte noch in verschiedener Hinsicht verbessert und ausgebaut werden. Ein weiterer Ausbau der Software bei unveränderter Hardware wäre durchaus möglich und könnte beispielsweise noch anspruchsvollere Lehrprogrammstrukturen oder noch eingehendere statistische Auswertungen erlauben. Durch eine leichte Modifikation der Tonsequenzmarkierung (Marke = Lücke im Dauerton) und der Tonbandelektronik könnte während des Suchprozesses eine Referenzgrösse für die Bandgeschwindigkeit gewonnen werden, die eine wesentlich bessere und zuverlässigere Regelung der Bandgeschwindigkeit erlauben würde, so dass der einwandfreie Bandtransport auch nach langem Lagern der Tonkassetten sicher gewährleistet wäre.

Moderne Kleincomputer sind heute viel leistungsfähiger als früher und werden oft mit den recht preisgünstigen Halbleiterspeichern ausgerüstet. Auch sind neuartige Massenspeicher (Ladungsgekoppelte Speicherelemente, Magnetblasenspeicher) in Entwicklung, die für Kleincomputer-CUU-Anwendungen gut geeignet wären. Zudem sind heute spezielle Video-Speicher mit beliebigem Zugriff erhältlich, die mit einfachsten Mitteln die Anzeige von computer-generiertem alphanumerischem Text auf einem Fernsehmonitor erlauben. Auch alphanumerische Tastaturen sind heute sehr preiswert. Es wäre also möglich, das Lehrsystem PLANETH mit relativ kleinem Aufwand so zu erweitern, dass es auch alphanumerischen Text verarbeiten könnte. Allerdings müsste zu diesem Zweck das Systemprogramm und das SESAM-Compilerprogramm wesentlich ausgebaut werden.

Die Konsolenelektronik könnte möglicherweise durch den Einsatz eines Mikroprozessors beträchtlich vereinfacht werden. Besonders geeignet wäre ein Mikroprozessor natürlich für die Steuerung der Diaprojektoren und des Tonbandgerätes. Dafür wäre aber noch einige Entwicklungsarbeit nötig, die sich nur lohnen würde, wenn der Aufwand in der Konsolenelektronik dadurch wesentlich gesenkt und eine grössere Anzahl Konsolen gebaut werden könnte.

- [11] W.Georgi u.a. Kleingruppenarbeit und computerunterstützter Unterricht in der Ingenieurausbildung. Heft 58 der Arbeitsgemeinschaft Hochschuldidaktik e.V., Hamburg, 1977
- [12] C.W.Churchman Operations Research - Eine Einführung in die Ackoff Unternehmensforschung. 5.Aufl., Oldenbourg Verlag, Arnoff München/Wien, 1971
- [13] H.Frank Kybernetische Grundlagen der Pädagogik. 2.Aufl., Agis Verlag, Baden-Baden, 1969 (2 Bände)
- [14] H.Ritter Das audiovisuelle Lehrsystem mit Kombikassette. Symposium Ingenieurpädagogik 1972, Klagenfurt. In : A.Melezinek (Hrsg.): Ergebnisse und Perspektiven der Ingenieurpädagogik, S. 225 ff., J.Heyn-Verlag, Klagenfurt, 1972
- [15] U.Lehnert Das Lehrsystem EDUCATOR. Zeitschrift für Datenverarbeitung, Band 10, S. 203 - 209, 1972
- [16] Simulation and Teaching Aids STA TM 150, die erste microcomputergesteuerte Inc. (STA) Lernmaschine mit eingebautem 150-Kanal Tonbandgerät, welche zwei Projektoren ansteuern kann. Prospekt der Firma STA, Stuketenstr. 11, CH-8332 Russikon/Schweiz, Feb. 1977
- [17] H.Müller Die Programmiersprache SESAM zur Lehranlage PLANETH. Institutsbericht 74-10 (mit Nachträgen), Mikrowellenlabor ETH Zürich, 1974
- [18] H.Müller Der SESAM-Compiler. Institutsbericht 74-11 (mit Nachträgen), Mikrowellenlabor ETH Zürich, 1975
- [19] B.F.Skinner The Science of Learning and the Art of Teaching. Harvard Educational Review 24, Nr. 2, S. 86 ff., 1954
- [20] P.Suppes The Uses of Computers in Education. Scientific American, Vol. 215, S. 207 ff., 1966

- [21] N.A.Crowder Automatic Tutoring by Intrinsic Programming. In: A.A.Lumsdaine/R.Glaser (Hrsg.) : Teaching Machines and Programmed Learning, a Source Book, S. 286 ff., National Education Association, Washington DC, 1960
- [22] N.A.Crowder Intrinsic and Extrinsic Programming. In: J.E.Coulson (Hrsg.) : Programmed Learning and Computer-Based Instruction, S. 58 ff., John Wiley & Sons, New York, 1962
- [23] H.Frank Kybernetische Pädagogik. In: B.S.Meder/W.Schmid (Hrsg): Kybernetische Pädagogik, Schriften 1958-1972, Band 2, S. 403 ff., W.Kohlhammer-Verlag, Stuttgart, 1973
- [24] K.Weltner Eine vergleichende Untersuchung von Lernleistung und Erinnerungsfestigkeit bei programmiertem Unterricht und Direktunterricht. In: H.Frank (Hrsg): Lehrmaschinen in kybernetischer und pädagogischer Sicht, Band 2, S. 11 ff., Klett-Verlag, Stuttgart 1964
- [25] H.Frank
E.Pietsch Rechner und programmierte Instruktion als pädagogische Werkzeuge. In: B.S.Meder/W.Schmid (Hrsg): Kybernetische Pädagogik, Schriften 1958-1972, Band 3, S. 204 ff., W.Kohlhammer-Verlag, Stuttgart, 1973
- [26] G.Epprecht Einsatz von Lehrmaschinen bei Hochschulkursen in Elektrotechnik. Symposium Ingenieurpädagogik 1975, Klagenfurt. In: A.Melezinek (Hrsg): Fortschritte der Ingenieurpädagogik, S. 173 ff., J.Heyn-Verlag, Klagenfurt, 1976
- [27] B.Stejskal UNITUTOR-Adaptiver Universal-Lehrautomat für kombinatorische Auswahl- und Freiwahlantworten. In: Praxis und Perspektiven des programmierten Unterrichts Band II, S.94 ff., Referate des 5.Symposiums über Lehrmaschinen, Verlag Schnelle, Quickborn, 1967

- [28] H.Freibichler Der Computer als Medium.
Online, Band 14, S. 540 ff., 1976
- [29] H.Freibichler Tendenzen in der Entwicklung und im Einsatz von
Systemen für den Computerunterstützten Unterricht
(CUU). Zeitschrift für Datenverarbeitung, Band 10,
S. 185 ff. & S. 314 ff., 1972
- [30] A.Manhart Description of Computer Controlled Teaching Machines.
Firmenschrift der SWISSAIR, Schweizerische Luft-
verkehrs AG, CH-8302 Kloten, Juni 1973
- [31] A.Schmitt Konzeption und Anwendung des RGU-Systems LEKTOR.
H.Hummel Angewandte Informatik, Band 18/5, S.199 ff., Mai 1976
- [32] K.J.Stetten TICCIT: A Delivery System Designed for Mass Utili-
sation. In: Proceedings of the 1972 Conference on
Cybernetics and Society, Oct 9-12, 1972, Washington
DC. IEEE Systems, Man and Cybernetics Society,
New York, 1972
- [33] C.R.Röhling GENIUS II - Ein System für den computergestützten
Unterricht. In: A.Melezinek (Hrsg): Ingenieur-
pädagogik und Computereinsatz, S. 299 ff.,
J.Heyn-Verlag, Klagenfurt, 1975
- [34] D.L.Bitzer PLATO - A Computer-Based System Used in the
R.L.Johnson Engineering of Education. Proceedings of the IEEE,
Vol. 59, No. 6, S. 960 ff., June 1971
- [35] L.F.Eastwood The PLATO IV CAI-System: Where is it now? Where can
R.J.Ballard it go? Journal of Educational Technology Systems,
Vol. 3, No. 4, S. 267 ff., Spring 1975
- [36] J.Nievergelt Interactive Systems for Education - the New Look of
CAI. In: O.Lecarme/R.Lewis (Hrsg): Computers in Edu-
cation, S.465 ff., IFIP, North Holland Publishing Co.,
1975

- [37] S.L.Pressey A Simple Apparatus which gives Tests and Scores and Teaches. School and Society Band 23, 1926. Nachgedruckt in : A.A.Lumsdaine/R.Glaser (Hrsg): Teaching Machines and Programmed Learning, a Source Book. Nat. Educ. Ass., Washington DC, 1960
- [38] K.Langeheinecke Der Stellenwert des didaktischen Mediums Computer im Ingenieurstudium. Symposion Ingenieurpädagogik 1972, Klagenfurt. In: A.Melezinek (Hrsg): Ergebnisse und Perspektiven der Ingenieurpädagogik, S. 191 - 195, J.Heyn-Verlag, Klagenfurt, 1972
- [39] M.Leibundgut Lernmaschine II. Semesterarbeit S71-4, Mikrowellenlabor ETH Zürich, 1971
- [40] F.Nyffeler Lernmaschine II. Semesterarbeit S71-5, Mikrowellenlabor ETH Zürich, 1971
- [41] H.Rööfli Lernmaschine II. Semesterarbeit S71-6, Mikrowellenlabor ETH Zürich, 1971
- [42] M.Bebawi Lernmaschine I. Semesterarbeit S71-1, Mikrowellenlabor ETH Zürich, 1971
- [43] R.Bührer Lernmaschine I, Programmierung. Semesterarbeit S71-2, Mikrowellenlabor ETHZ, 1971
- [44] H.Strassmann Lernmaschine I, Konsolenlogik. Semesterarbeit S71-3, Mikrowellenlabor ETHZ, 1971
- [45] E.Kaganas Digital adressierbares Kassettengerät für Sprachwiedergabe. Semesterarbeit S73-6, Mikrowellenlabor ETH Zürich, 1973
- [46] E.Kaganas Tonbandgerät für Lehrmaschine PLANETH. Diplomarbeit D73-2, Mikrowellenlabor ETH Zürich, 1973
- [47] R.Cattin Steuerelektronik zu PLANETH-Tonbandgerät. Semesterarbeit S74-1, Mikrowellenlabor ETH Zürich, 1974

- [48] R.Cattin Elektronik in CMOS-Technik für Tonbandgerät zu PLANETH-Lehrsystem. Diplomarbeit D74-3, Mikrowellenlabor ETHZ, 1974
- [49] U.Waber Systembeschreibung PLANETH 1. Institutsbericht 73-15, Mikrowellenlabor ETH Zürich, 1973
- [50] U.Waber Programmierung und Bedienung der PLANETH-1-Konsole. Institutsbericht 73-16, Mikrowellenlabor ETH Zürich, 1973
- [51] G.Epprecht Vorlesungskurs mit elektronischem Rückmeldesystem. In: A.Melezinek (Hrsg): Die Technik und ihre Lehre, S. 261 ff., J.Heyn-Verlag, Klagenfurt, 1974
- [52] H.Häberlin Uebersicht über das computergesteuerte Lehrsystem
G.Epprecht PLANETH. In:(Hrsg): Moderne Methoden und Hilfsmittel in der Ingenieurausbildung, ... , Darmstadt (?), 1978 (in Vorbereitung)
- [53] W.Rappaport Tailor-Made Teaching Through TICCIT. The MITRE Matrix,
E.Olenbush Vol. 8, No. 4, The MITRE Corp., Bedford, Mass., USA , 1975
- [54] D.Alpert Advances in Computer-Based Education.
D.L.Bitzer Science, Vol. 167, S. 1582 ff., 1970
- [55] D.Alpert The Design of an Economically Viable Large-Scale Computer-Based Education System. CERL Report X-5, Computer-Based Education Research Laboratory, University of Illinois, USA, Jan 1972
- [56] Schweizerische Rapport du Groupe ad hoc chargé d'étudier la question de
Koordinations- l'Enseignement Assisté par Ordinateur à la Division de la
Stelle für Science et de la Recherche du Département fédéral de
Bildungs- l'Intérieur. Centre suisse de coordination pour la
forschung recherche en matière d'éducation, Aarau, 1972
- [57] R.T.Murphy Evaluation of the PLATO IV Computer-Based Education
L.R.Appel System in the Community College.
ACM SIGCUE Bulletin Vol. 12, No. 1, Jan 1978
- [58] The MITRE An Overview of the TICCIT Program.
Corporation MITRE publication M74-1, The MITRE Corp., Jan 1974

LEBENS LAUF

Ich wurde am 9. Februar 1947 in Basel geboren. Nach der Primarschule in Muttenz besuchte ich während acht Jahren das Realgymnasium in Basel, wo ich im Frühling 1966 die Maturitätsprüfung Typ B ablegte. Anschliessend studierte ich während eines Semesters Physik an der Universität Basel. Nach einigen Monaten Praxis immatrikulierte ich mich im Herbst 1966 an der Eidgenössischen Technischen Hochschule in Zürich. Mein Studium an der Abteilung für Elektrotechnik (IIIB) schloss ich im Juni 1971 mit dem Diplom als Elektroingenieur ab. Im Sommer 1971 trat ich eine Assistentenstelle am Mikrowellenlabor der ETH an, wo diese Arbeit entstand.