

Bedienungsanleitung

DATA I/O

System 19
Universal Programmer



MACROIRON 

Cosimastraße 4, 8000 München 81, Tel. 089/915061

I N H A L T
=====

Universal-Programmer SYSTEM 19 von Data I/O

<u>Kap.</u>	<u>Titel</u>	<u>Seite</u>
1.	Kurzbeschreibung des SYSTEM 19	1
1.1	Spezifikation des SYSTEM 19	2
1.2	Hexadezimalen Zahlensystem	2
1.3	Betriebshinweise für das SYSTEM 19	4
1.3.1	Programmiermodule	4
1.3.2	Socket-Adapter	5
1.4	Serieller Interface-Anschluß	5
1.4.1	Interface Pin-Belegung	7
2.	Bedienungsanleitung	10
2.1	Einführung	10
2.1.1	Programmer-RAM	11
2.1.2	Beginn- und Endadresse des RAM	12
2.1.3	SUM-CHECK (Prüfsummenbildung)	12
2.1.4	Nachprüfung des internen RAM	13
2.1.5	Fehlererkennung	13
2.2	Betriebshinweise	13
2.2.1	Einschalten	13
2.2.2	Programmierskarte und Socket	14
2.2.3	PROM	14
2.3	Umgang mit PROMs	14
2.3.1	Betriebsart	14
2.3.2	Setzen oder Überprüfen der Anfangsadresse	17
2.3.3	Setzen oder Überprüfen der Endadresse	17
2.3.4	Laden eines Master PROM in das RAM	17
2.3.5	Programmieren eines PROM	18
2.3.6	PROM Überprüfung	19
2.3.7	DEVICE-DATA-Taste	19
2.4	Dateneingabe über das Tastenfeld	20
2.4.1	Tastefeldanwahl und Adressenauswahl	20
2.4.2	Übertragung der Daten vom Tastenfeld zum internen RAM	20
2.4.3	EDIT-Funktion	21
2.5	SELECT-CODE-Operationen	22
2.5.1	Setz- und Abfrage-Befehle	24
2.5.2	Serielle Interface-Operationsbefehle	25
3.	Translator-Spezifikation	26
3.1	Datenwörter	27
3.2	Start und Ende	27
3.3	Adressenfeld	27
3.4	Prüfsumme	27
3.5	Eingangs-Charakterisierung	28
3.6	Ausgangs-Charakterisierung	28

INHALT - FORTSETZUNG

=====

<u>Kap.</u>	<u>Titel</u>	<u>Seite</u>
4.	Befehle für die Datenübertragung	29
4.1	Vorbereitung	29
4.1.1	Polarität	29
4.1.2	Kompatibilität von PARITY, STOP-Bit und Baud-Rate	30
4.1.3	Wortbreite	30
4.1.4	Leer- und Startbedingung	30
4.1.5	Start- und Endcode	30
<u>Anhang</u>		
<u>Programmierbare Halbleiter-Speicher</u>		
1.	Programmiertechnologien	31
1.1	Festwertspeicher nach dem Fusible-Link-Verfahren	31
1.2	Festwertspeicher nach AIM-Verfahren	33
1.3	UV-Löschbare Festwertspeicher	34
1.4	Elektrisch löschbare Festwertspeicher EAROMs	35
2.	Programmierbare Logik-Bausteine	36
2.1	Programmierbare Festwertspeicher (PROM-Programmable Read-Only-Memory)	36
2.2	Programmierbare Dioden-Matrix	38
2.3	Programmierbare Multiplexer (PMUX Programmable Multiplexer)	39
2.4	Programmierbare Gatter-Anordnung (FPGA Field Programmable Gate Array)	40
2.5	Programmierbare Array-Logik (PAL Programmable Array Logic)	41
2.6	Programmierbare Logik-Anordnung (FPLA Field Programmable Logic Array)	42
2.7	Handhabung von MOS-PROMs	43
2.8	Löscheinrichtungen für EPROMs	43
2.9	EPROM-Löschtest	44
	Anhang A	47

Universal-Programmer SYSTEM 19 von Data I/O

1. Kurzbeschreibung des SYSTEM 19

Unter der Bezeichnung "SYSTEM 19" hat Data I/O einen neuen Universal-Programmer mit einem Mikroprozessor entwickelt. Bei der Entwicklung dieses Systems ist Data I/O davon ausgegangen, daß mit der Anwendung immer vielseitiger programmierbarer Speicherelemente wie

PROMs, EPROMs, EAPROMs, VEPROMs usw.
Dioden-Matrizen,
Programmierbare Multiplexer,
Freiprogrammierbare Gatter-Anordnungen,
Freiprogrammierbare Logik-Anordnungen und
Programmierbare Anordnungs-Logik

auch die Anforderungen an ein geeignetes Programmiergerät erhöht werden müssen. Außerdem werden in den nächsten Jahren weitere Speicherentwicklungen erwartet, die sich durch Technologie und Terminologie von den heutigen Typen unterscheiden werden. Die Zuordnung der einzelnen Speicherfamilien wird bei dem SYSTEM 19 durch einzelne Programmiermodule erreicht. Diese Karten sind für alle Data I/O-Geräte (auch bei älteren Programmern) verwendbar. Neu bei dem SYSTEM 19 ist die sehr leichte Austauschbarkeit dieser Programmkarten, ohne daß ein Datenverlust auftritt. Auf diese Weise kann ein identisches Programm in mehrere Speicherelemente verschiedener Technologien eingeschrieben werden.

Hardware-orientierte Programmkarten sind nach wie vor die einfachste und wirtschaftlichste Lösung, um die von den PROM-Herstellern vorgeschriebenen Programmierspezifikationen genau einhalten zu können.

Alle führenden Hersteller von programmierbarer Logik haben weltweit die Data I/O-Programmiermodule freigegeben.

Die Data I/O-Programmiergeräte werden weltweit eingesetzt von folgenden Firmen:

AEG, AT & T, Beckman, British Aircraft, Burroughs, Chrysler, DEC, Ferrantie, Ford, GE, GM, Grumman, H-P, Hitachi, Honeywell, IBM, GEC-Ltd, Fluke, Marconi-Elliott, Nixdorf, Olivetti, Philips, Polaroid, Raytheon, Saab-Scania, Sharp, Siemens, Tektronix, Thompson und Westinghouse. + LITEF

Eine weitere Neuerung bei dem SYSTEM 19 besteht aus der in zwei Funktionsebenen aufgeteilten Systemsoftware. Die erste Ebene erlaubt die unkomplizierte Bedienung für Programmier- und Edit-Funktionen. Hier ist der Programmierer von der Eingabe der Anfangs- und Endadresse befreit, wie dies auch bei anderen Programmern üblich ist. Die zweite Ebene ist für alle speziellen Anforderungen vorgesehen, wobei die Anwahl über eine SELECT-CODE und BEGIN-RAM-Taste erfolgt. Mit der SELECT-CODE-Anwahl werden folgende Sonderbefehle ausgeführt:

High- und Low-Order des Datenbytes vertauschen
internes RAM mit einer Variablen auffüllen,
Prüfsumme (Chek-Sum) bilden

RAM-Inhalt invertieren
gewünschtes Datenformat und die I/O-Schnittstelle anwählen
Kalibrierungszusatz anwählen.

Die zusätzliche BEGIN-RAM-Taste erlaubt ein Laden und Abrufen von Daten in beliebigen Adressenbereichen.

Die Frage nach einem getrennten Sockel für das Master-PROM wird durch ein zusätzliches automatisches Testprogramm, die eine versehentliche Fehlprogrammierung bereits programmierter PROMs ausschließt, beantwortet. So werden neben anderen Tests, die vor dem eigentlichen Programmiervorgang durchlaufen werden, ein ILLEGAL-Bit-Test durchgeführt, der bei einem negativen Testergebnis eine Programmierung verhindert. Bei diesem Test wird der RAM-Inhalt mit dem zu programmierenden PROM überprüft. Zusätzliche Bit im RAM werden automatisch akzeptiert, jedoch Leerstellen im RAM, bei denen im PROM an gleicher Stelle Bit programmiert sind, führen zu einem negativen Ergebnis. Dieser Vorgang des Überprogrammierens ist besonders in der Entwicklung von Software-Programmen wichtig.

Der interne RAM-Datenspeicher hat eine Standard Speicherkapazität von 32k-Bit und kann bis zu 128k-Bit erweitert werden. Dies ist nicht nur ein besonderes Zugeständnis für zukünftige PROM-Typen, sondern der gesamte RAM-Speicher kann mit Daten voll belegt werden und beim Programmieren in beliebigen Teilen aufgerufen werden. Damit hat der Entwickler oder Programmierer die Wahl, vom vollen Adressenbereich in einen Teiladressenbereich überzugehen, ohne daß Daten verloren gehen. Ein noch wesentlicher Betrag zur Programmsicherheit ist ein automatischer Geräte-Selbsttest und eine akustische Bedienungsunterstützung.

Achtung ! Vor dem Ein- und Ausschalten des SYSTEM 19 sowie beim Auswechseln des Program-Paks darf sich kein PROM in dem Sockel befinden.

1.1 Spezifikation des SYSTEM 19

Spannungsbereiche: 100 V, 120 V, 220 V und 240 V Wechselfspannung (50/60 Hz). Das Gerät ist auf 220 V eingestellt.

Leistungsverbrauch : 35 W

Mechanische Gehäuseabmessungen: 38,1 cm x 15,2 cm x 15,3 cm

Gewicht: 6,4 kg

Betriebstemperatur: 0 bis 45 °C

Lagertemperatur: -40 bis 55 °C

1.2 Hexadezimaler Zahlensystem

Die Dateneingabe in das SYSTEM 19 erfolgt im Hexadezimalen Zahlensystem. Dieses Zahlensystem hat im Gegensatz zum Dezimalsystem (Z_{10}) die Basis von $B = 16$. Für viele Anwendungen in der Elektronik gilt das binäre Zahlensystem mit der Basis von $B = 2$. Daher werden die einzelnen Daten umgerechnet:

Dezimalsystem	Binärsystem	Hexadezimalsystem
0	0 0 0 0	0
1	0 0 0 1	1
2	0 0 1 0	2
3	0 0 1 1	3
4	0 1 0 0	4
5	0 1 0 1	5
6	0 1 1 0	6
7	0 1 1 1	7
8	1 0 0 0	8
9	1 0 0 1	9
10	1 0 1 0	A
11	1 0 1 1	B
12	1 1 0 0	C
13	1 1 0 1	D
14	1 1 1 0	E
15	1 1 1 1	F

Die Umwandlung von dem Dezimalsystem in das Hexadezimalsystem erfolgt durch eine Division mit der Zahl 16:

$$B_{10} = 92 \quad B_{16} = ?$$

$$\begin{array}{r} 92 : 16 = 5 \text{ Rest } \boxed{C} \\ 5 : 16 = 0 \text{ Rest } \boxed{5} \end{array}$$

Leserichtung

Es ergibt sich die Zahl 5C.

Die Umwandlung von dem Dezimalsystem in das Binärsystem erfolgt durch eine Division mit der Zahl 2:

$$B_{10} = 92 \quad B_2 = ?$$

$$\begin{array}{r} 92 : 2 = 46 \text{ Rest } \boxed{0} \\ 46 : 2 = 23 \text{ Rest } \boxed{0} \\ 23 : 2 = 11 \text{ Rest } \boxed{1} \\ 11 : 2 = 5 \text{ Rest } \boxed{1} \\ 5 : 2 = 2 \text{ Rest } \boxed{1} \\ 2 : 2 = 1 \text{ Rest } \boxed{0} \\ 1 : 2 = 0 \text{ Rest } \boxed{1} \end{array}$$

Leserichtung

Es ergibt sich die Zahl 1011100.

Tetradenschreibweise:

$$\underbrace{0101}_5 \quad \underbrace{1100}_C$$

Die Eingabe der Adressen erfolgt ebenfalls in Hexadezimaler Form:

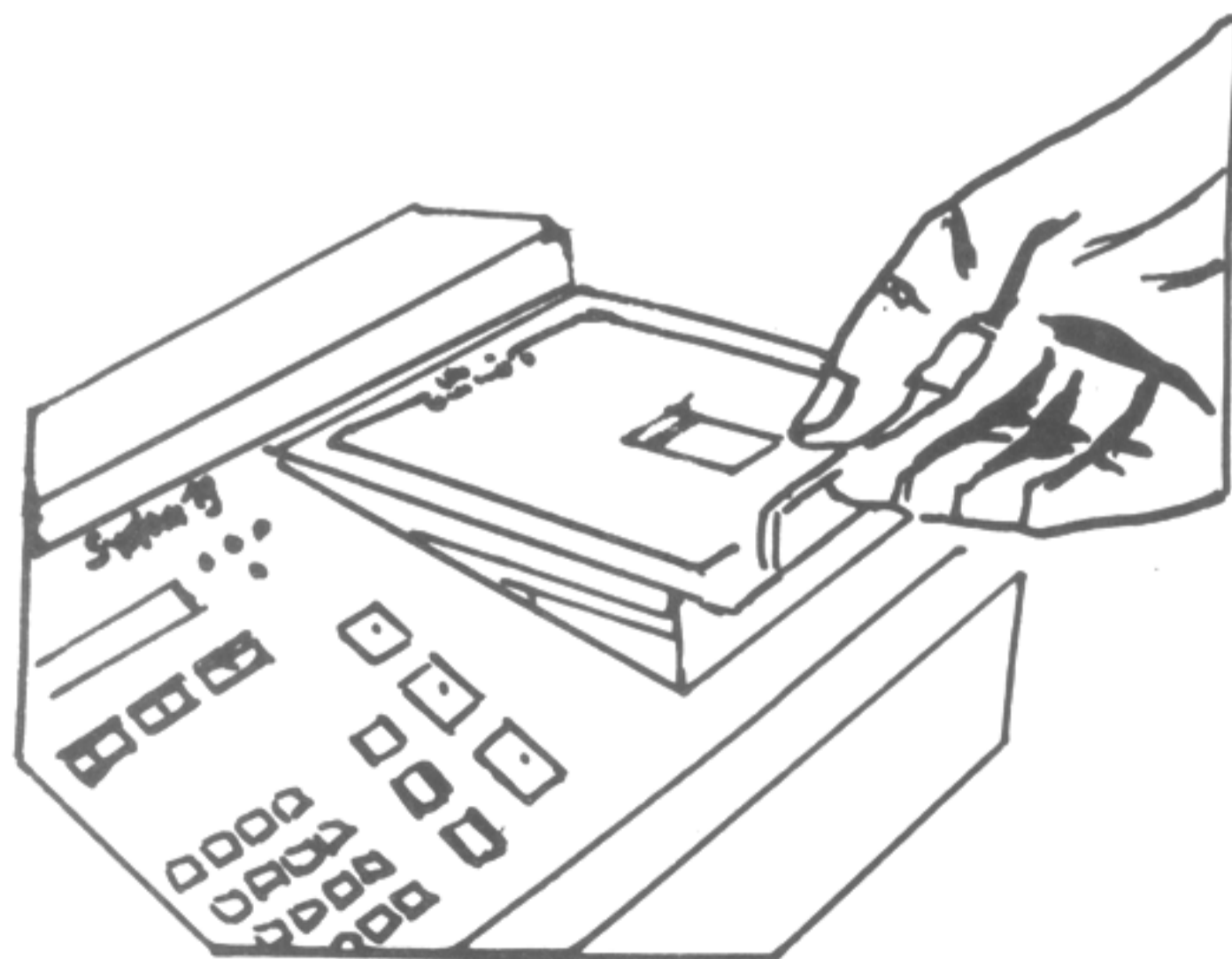
16^0	=	1	=	1
16^1	=	16	=	10
16^2	=	256	=	100
16^3	=	4 096	=	1 000
16^4	=	65 536	=	10 000
16^5	=	1 048 576	=	100 000
16^6	=	16 777 216	=	1 000 000

1.3 Betriebshinweise für das SYSTEM 19

Das SYSTEM 19 wird bereits von der Firma MACROTRON auf die richtige Betriebsspannung für den europäischen Raum eingestellt. Für den Anwender gelten daher nur folgende Hinweise:

1.3.1 Programmiermodule

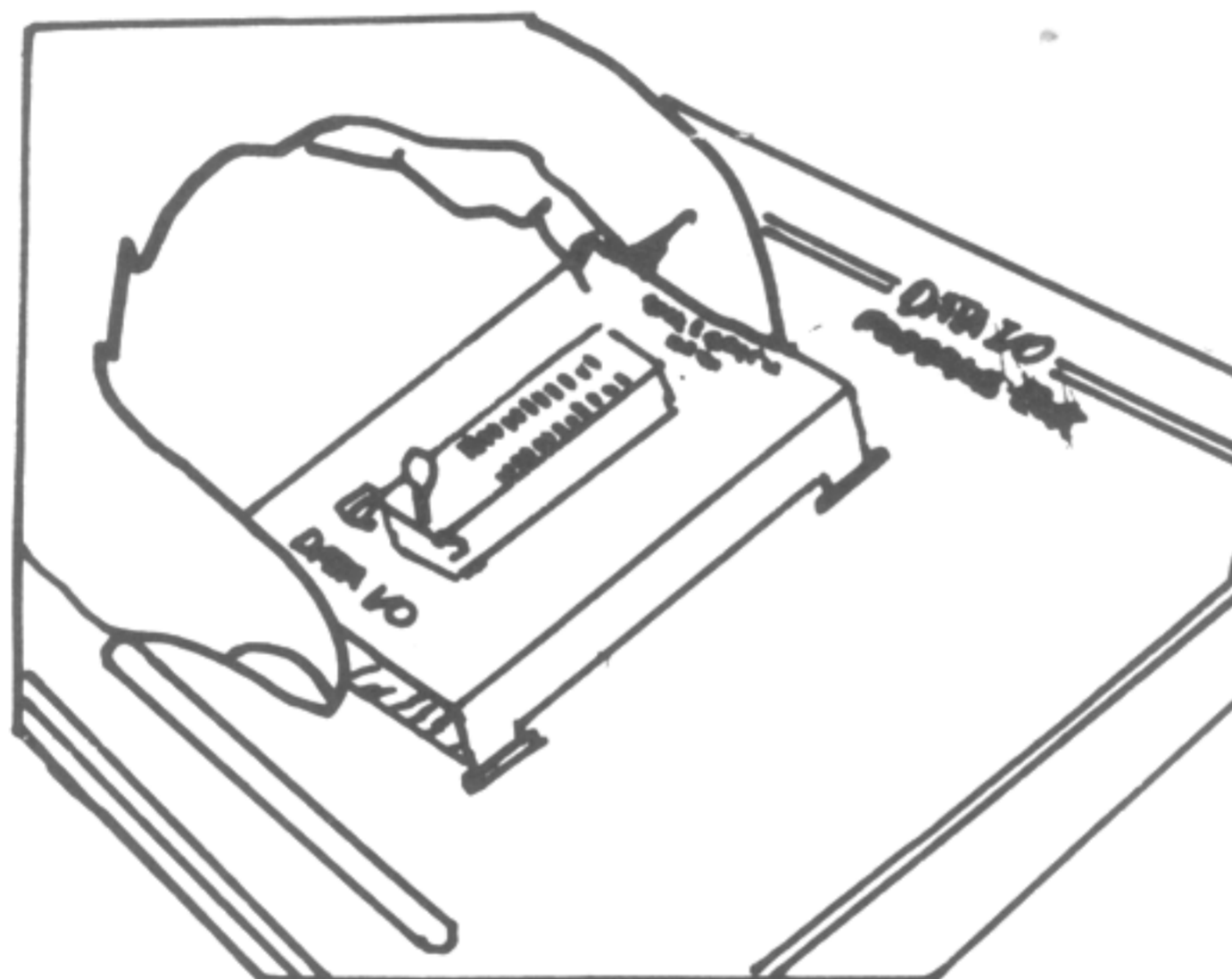
Für die einzelnen Speicherelemente bzw. PROM-Familien werden unterschiedliche Programmiermodule verwendet, die für das betreffende Speicherelement gültig sein müssen. Bild 1.1 zeigt das Einschieben der Programmierkarte in das SYSTEM 19. Bei dem Modulwechsel darf die Betriebsspannung eingeschaltet sein, damit beim Wechseln kein Datenverlust in dem internen RAM auftreten kann.



Das SYSTEM 19 setzt beim Einschalten der Betriebsspannung automatisch den RAM-Datenspeicher auf 0 zurück. Der Netzschalter befindet sich an der Rückseite des Programmers. Beim Wechseln der Programmierkarten schalten sich die Anzeigen aus und es wird ein akustisches Alarmzeichen gegeben. Erst wenn die Karte richtig in der Steckerleiste ist, verstummt der Warnton.

1.3.2 Socket-Adapter

Der Socket-Adapter für die Bauelemente wird auf das Programmiermodul gesteckt, wie dies Bild 1.2 zeigt. Dieser Socket kann nicht verkehrt gesteckt werden, da eine mechanische Sicherung vorhanden ist. Der Socket stellt die richtige Pin-Zuordnung zwischen Programmiermodul und dem zu programmierenden PROM her.



1.4 Serieller Interface-Anschluß

Das SYSTEM 19 kann durch ein RS232-Interface mit verschiedenen Datenformaten angesteuert werden. Auf der Rückseite des Gerätes befindet

sich ein kompatibler Anschluß für Computer und Terminals 20mA/V24.
Bild 1.3 zeigt den Aufbau dieser Option.

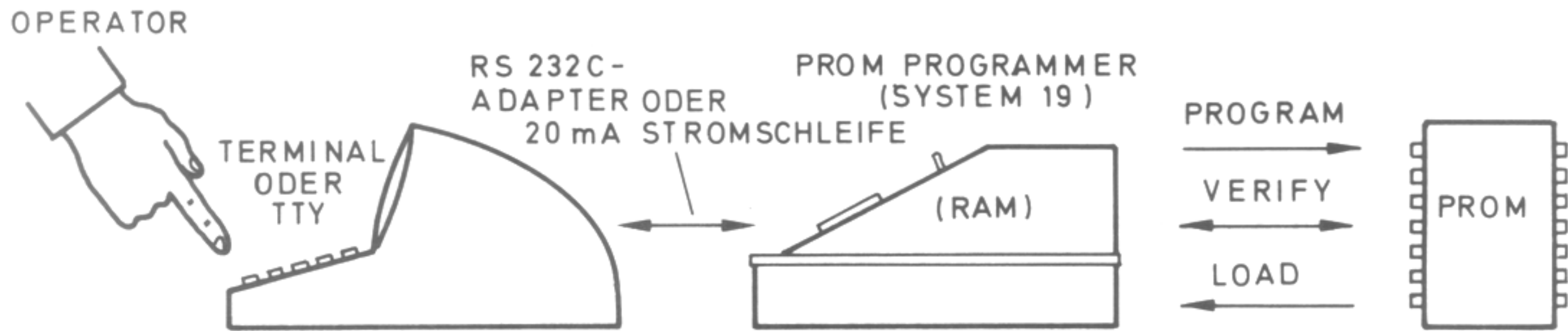


Bild 1.3 SYSTEM 19 in Verbindung mit peripheren Einheiten

Folgende Datenformate sind möglich:

- a) 990-1900 , Einzelformat: 50 Hex-ASCII-Space
- b) 990-1901 , Mehrfachformat mit gesamt 21 unterschiedlichen Translatorformaten:

- 50 Hex-ASCII-Space
- 01(05) ASCII-BNPF
- 02(06) ASCII-BHLF
- 03(07) ASCII-BIOF
- 08(09) 5 Level BNPF
- 10 Binary
- 11 DEC Binary
- 30 ASCII-Octal (Space)
- 31 ASCII-Octal (Percent)
- 32 ASCII-Octal (Apostrophe)
- 51 ASCII-Hex (Percent)
- 52 ASCII-Hex SMS (Apostrophe)
- 53 ASCII-Hex (Comma)
- 70 RCA Cosmac
- 71 Intel Non-Intellec
- 80 Fairchild F8 Formulator
- 81 MOS Technology Development System
- 82 Motorola Exerciser
- 83 Intel MDS und Intellec 8
- 85 Signetics Absolute Object
- 86 TEK HEX
- () ohne Startzeichen STX

- c) 990-1902 , beinhaltet b) F1 Computer-Remote-Control mit Einzelformat:

- 50 Hex-ASCII-Space
- 01(05) ASCII-BNPF
- 02(06) ASCII-BHLF
- 03(07) ASCII-BIOF
- 08(09) 5 Level BNPF

10	Binary
11	DEC Binary
30	ASCII-Octal (Space)
31	ASCII-Octal (Percent)
32	ASCII-Octal (Apostrophe)
51	ASCII-Hex (Percent)
52	ASCII-Hex SMS (Apostrophe)
53	ASCII-Hex (Comma)
70	RCA Cosma
71	Intel Non-Intellec
80	Fairchild F8 Formulator
81	MOS Technology Development System
82	Motorola Exerciser
83	Intel MDS and Intellec 8
85	Signetics Absolute Object
86	TEK HEX

d) 990-1930 , beinhaltet b) F1 Terminal-Remote-Control mit 21 Translatorformaten:

50	Hex-ASCII-Space
01(05)	ASCII-BNPF
02(06)	ASCII-BHLF
03(07)	ASCII-BIOF
08(09)	5 Level BNPF
10	Binary
11	DEC Binary
30	ASCII-Octal (Space)
31	ASCII-Octal (Space)
32	ASCII-Octal (Apostrophe)
51	ASCII-Hex (Percent)
52	ASCII-Hex SMS (Apostrophe)
53	ASCII-Hex (Comma)
70	RCA Cosmac
71	Intel Non-Intellec
80	Fairchild F8 Formulator
81	MOS Technology Development System
82	Motorola Exerciser
83	Intel MDS and Intellec 8
85	Signetics Absolute Object
86	TEK HEX

Durch diese Datenformate ist ein optimaler Anschluß zu Computer, Datensichtgeräten, Teletype und Leser möglich.

1.4.1 Interface Pin-Belegung

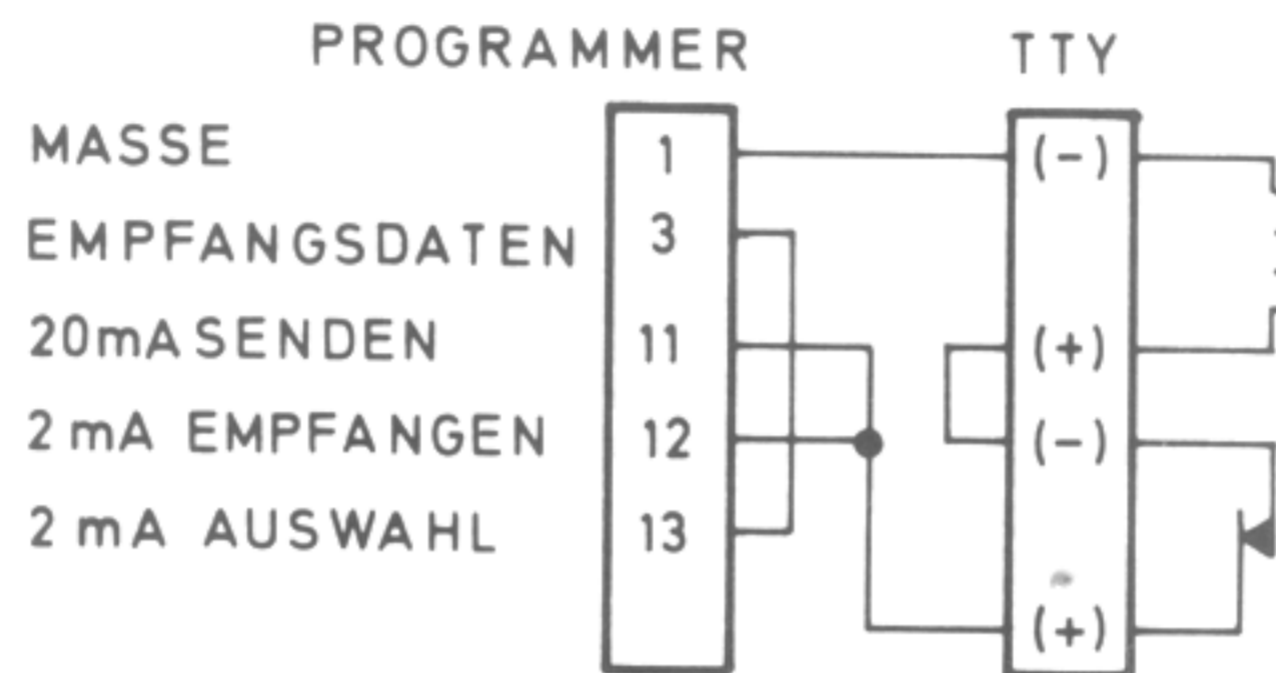
Die nachfolgende Tabelle zeigt die Pinbelegung für das Interface:

Pin	Signalbedeutung	Erklärung
1	Masse	Durch diese Verbindung zwischen Programmierer und Interface ergibt sich gleiches Bezugspotential und daher ein kompatibles System. Bei TTY Anschluß ist dies die Rückkopplung für das 12 VDC-Signal vom TTY Terminal.

Pin	Signalbedeutung	Erklärung
2	Transmit Data	Übertragungsleitung zum Senden von Daten- und Kontrollzeichen vom SYSTEM 19 (Spannungspegel +12 V und -5 V). RS232C = V24
3	Receive Data	Übertragungsleitung des SYSTEM 19 um Daten und Kontrollzeichen zu empfangen. (RS232C = V24)
4	Request to Send	Diese Leitung hat normalerweise einen H-Pegel durch den Programmer. Bei einem L-Pegel wird die Datenübertragung vom Terminal in den Programmer blockiert.
5	Clear to Send	Über diese Leitung kontrolliert das Terminal den Datentransfer zum M17/19. Ein H-Pegel erlaubt eine Datenübertragung, ein L-Pegel blockiert sie.
6	Data Set Ready	Ist im M17/19 intern mit P20 verbunden (siehe P20).
7	Signal-Masse	Stellt die Signalmassenverbindung zwischen dem Programmer und dem RS232C-Datenterminal her.
8	Carrier Detect	Diese Leitung wird vom Terminal als H-Pegel geschaltet, wenn das Terminal an das Modell 17/19 angeschlossen ist. Ein L-Pegel blockiert das I/O vom M17/19.
9	+24VDC	Anschluß für externen Gebrauch
10		nicht angeschlossen
11	20mA-Send	Datenübertragung mit einer 10mA-Stromschleife für einen TTY-Anschluß.
12	2mA-Receive	Dateneingang für 2mA Stromschleifen für einen TTY-Anschluß.
13	2mA-Detect	Datenausgang nach Umwandlung von 2mA auf RS232C = V24. Bei TTY-Betrieb ist P13 mit P3 (Receive Data RS232C = V24) zu verbinden.
14		nicht angeschlossen
15		Reserve für zukünftige Entwicklungen
16-19		nicht angeschlossen
20	Data Ready	Ist im M17/19 intern mit P6 verbunden. Ein H-Pegel an P20 zeigt an, daß das Terminal bereit ist. An P6 wo das Signal von P20 durchgeschleift ist, zeigt es die Bereitschaft des Programmierens.
21		nicht angeschlossen
22	+5VDC	Anschluß für externen Gebrauch

Pin	Signalbedeutung	Erklärung
23	-5VDC	Anschluß für externen Gebrauch
24		nicht angeschlossen
25		nicht angeschlossen

Bild 1.4 zeigt die Anschlußmethoden für serielle Interface-Schaltungen. Das SYSTEM 19 ist ein ungebuffertes System und benötigt für den Datentransfer eine Datenrate, Geschwindigkeit, die entsprechend eingestellt werden muß. Hierzu dient der Drehschalter an der Rückseite des Gerätes, wie in Bild 1.6 gezeigt wird. Ebenso gehören hierzu die Schalter für die einzelnen Statusfunktionen.



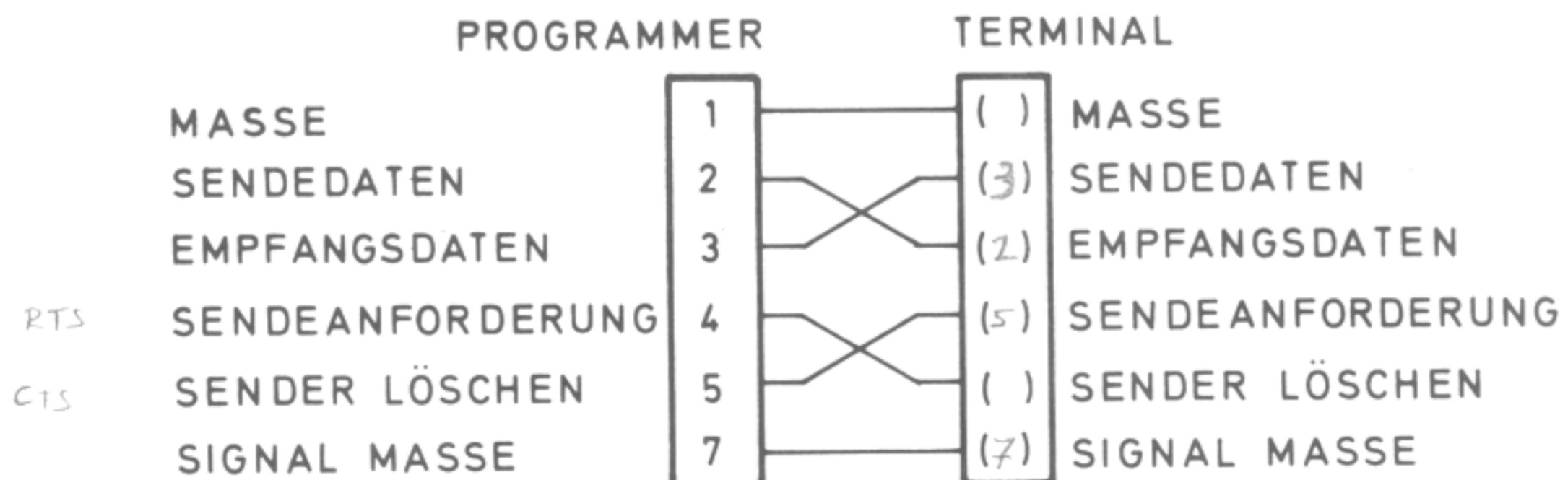
20 mA-Schleifenstrom-Verbindung, halber Duplex-Betrieb



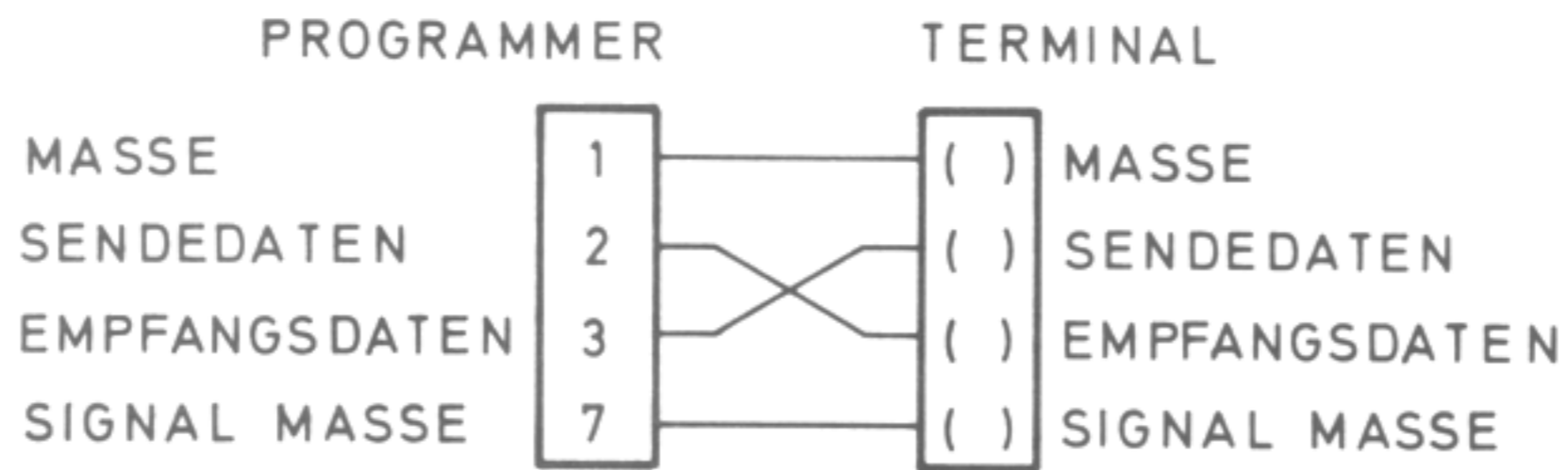
20 mA-Schleifenstrom-Verbindung, voller Duplex-Betrieb

Bild 1.4 Anschlußmethoden für serielle Interface-Schaltungen

System 29/05



a) RS232-Anschlußschema, halber/voller Duplex-Betrieb mit "handshake"



b) RS232-Anschlußschema, halber/voller Duplex-Betrieb ohne "handshake"

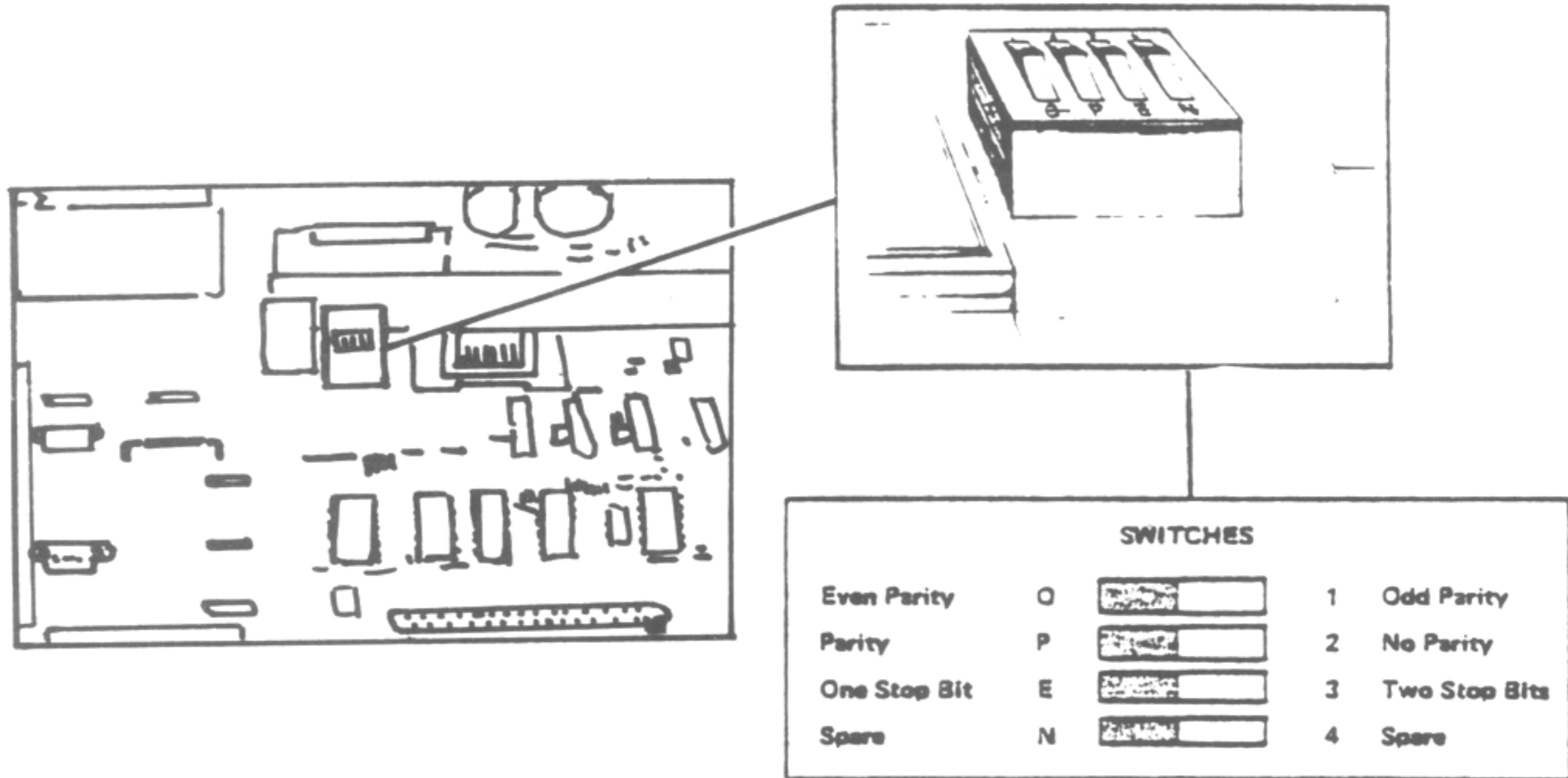


Bild 1.5 I/O Status- Schalter für Parity und Stop Bit

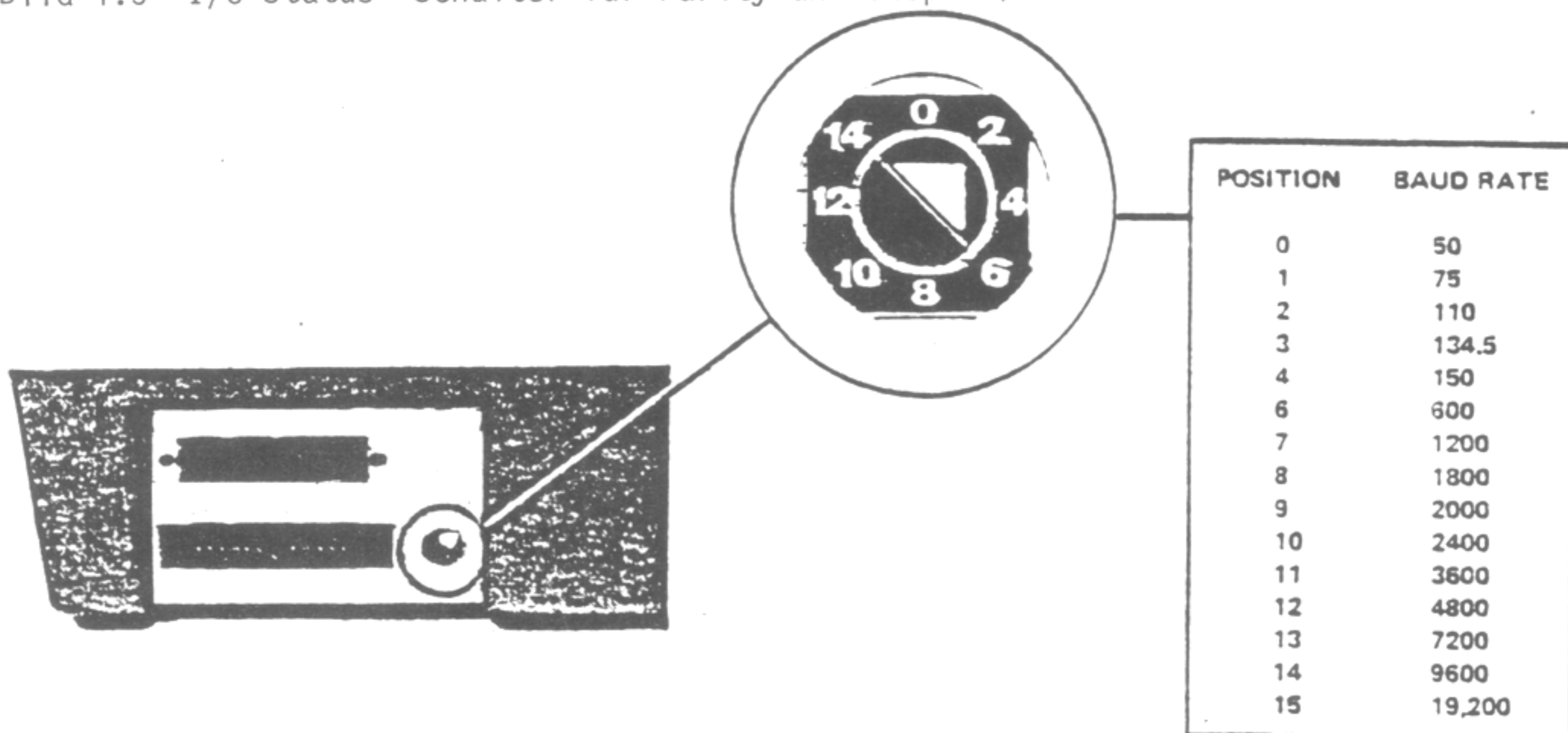


Bild 1.6 Baud-Rate Selector

2. Bedienungsanleitung

2.1 Einführung

Dieser Abschnitt zeigt die Möglichkeiten für die Programmierung von Logikbausteinen durch das SYSTEM 19. Hierzu gehört die Datenmanipulation, die verschiedenen Speicheroperationen und die Programmierfunktion in Verbindung mit dem internen RAM. Die Daten für einen Programmiervorgang werden in das interne RAM geladen. Eine direkte Programmierung eines PROMs ist bei dem SYSTEM 19 aus Gründen der Sicherheit nicht beabsichtigt, da bei einer direkten Programmierung zusätzliche Übertragungsfehler auftreten können. Außerdem sind bei den meisten Speichertechnologien mehrere Programmierschleifen oder Funktionszyklen notwendig. Der RAM-Datenspeicher des SYSTEM 19 wird entweder von einem Master-PROM (programmierter Festwertspeicher), von dem Bedienungsfeld oder von dem seriellen Dateneingang geladen. Dadurch ergeben sich für den Anwender verschiedene Formate für den Ladebetrieb des internen RAM-Datenspeichers.

Das SYSTEM 19 ist in der Lage laufend Kontrollfunktionen durchführen zu können. Wenn man das Anzeigenfeld und das Tastenfeld für die Eingabe von Informationen verwendet, ergeben sich drei Möglichkeiten:

- a) Der Befehl bezieht sich auf das Programmieren, Laden und Vergleichen
- b) Die Tastenfeld-Operation wie z.B. Datenein- und ausgabe, Datenwechsel
- c) Ein- und Ausgabeoperationen mit der seriellen Schnittstelle, Datenformatspezifikationen und Datenmanipulationen.

Bei einem typischen Programmablauf können Teile von diesen Operationsgruppen verwendet werden. Das Programm eines Master-PROM kann in das RAM geladen werden und mit Hilfe des Tastenfeldes läßt sich der Inhalt des RAM entsprechend verändern. Von dem geänderten RAM-Inhalt läßt sich ein Lochstreifen oder eine Lochkarte erstellen und anschließend wird dieses neue Programm in ein PROM programmiert.

Die nachfolgenden Abschnitte beschreiben die Arbeitsweise des SYSTEM 19. In dem Bild 2.1 sind die einzelnen Funktionen des Bedienungsfeldes gezeigt.

Alle Datenübertragungen oder Vergleichsoperationen zwischen dem RAM-Inhalt und dem seriellen Dateneingang oder dem Inhalt des Master-PROM können einfach durchgeführt werden. Die nachfolgende Tabelle definiert die einzelnen Operationseigenschaften:

- | | |
|----------------------------|--|
| a) LOAD-Funktion: | Datentransport vom Master-PROM in das RAM |
| b) VERIFY-Funktion: | Datenvergleich zwischen PROM und RAM |
| c) PROGRAM-Funktion: | Datenübertragung vom RAM in das PROM |
| d) INPUT-Funktion: | Datenübertragung vom seriellen Eingang in das RAM |
| e) INPUT-COMPARE-Funktion: | Datenvergleich zwischen serielltem Eingang und RAM |
| f) INPUT-COMPARE-Funktion: | Datenübertragung vom RAM zum seriellen Ausgang |

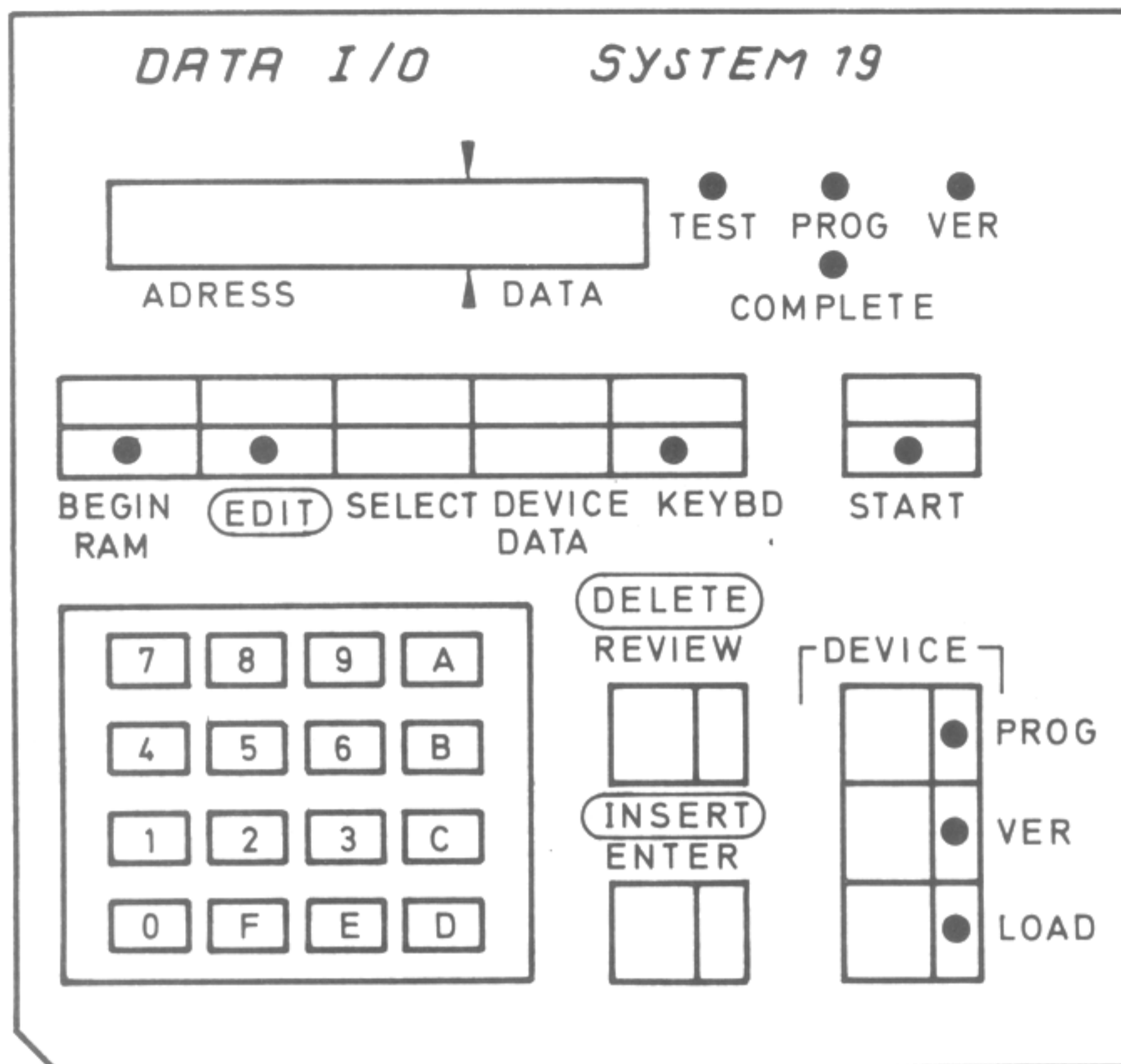


Bild 2.1 Bedienungsfield des SYSTEM 19

2.1.1 Programmer-RAM

Das SYSTEM 19 hat in der Standard-Ausführung ein 32k-RAM mit einer Organisation von 4096 x 8 Bit. Damit hat dieser RAM-Datenspeicher einen Adressbereich zwischen der Anfangsadresse 0000 und der Endadresse 0FFF. Die Auswahl der Adressen erfolgt im hexadezimalen (se-dezimal) Zahlensystem über das Tastenfeld. Durch zusätzliche RAM-Speicherplätze kann das SYSTEM 19 auf 128k erweitert werden. Dadurch ergibt sich ein Adressbereich zwischen 0000 und 3FFF. Die Masteradresse beginnt normalerweise bei der Adresse 0. Die Endadresse wird von dem Sockeladapter voreingestellt.

Das SYSTEM 19 ist für eine Wortbreite zwischen 4- und 8-Bit ausgelegt. Das interne RAM paßt sich der charakteristischen Wortbreite der Programmierspeicherelemente automatisch durch die einzelnen Programmierkarten an. Bei einem 4-Bit-Ausgang arbeitet das SYSTEM 19 mit nur 4-Bit für die Datenprogrammierung. Dies gilt dann auch für die Manipulation. Eingabe und Löschen der Daten in dem RAM. Der übrige RAM-Speicher mit 3k x 4 kann als zusätzlicher Speicher verwendet werden. Er wird beispielsweise benötigt, um von der 8-Bit-Form in ein 4-Bit-Datenwort überzugehen.

2.1.2 Beginn- und Endadresse des RAM

Das SYSTEM 19 ist an keine Anfangs- und Endadresse gebunden. Der Wert der Beginnadresse kann durch die Begin-RAM-Taste festgelegt werden und dadurch kann festgelegt werden, bei welcher RAM-Adresse die Datenübertragung starten soll. Wird keine Beginnadresse festgelegt, beginnt die Datenübertragung immer bei der Adresse 0000. Mit dem Ladebefehl wird die Adresse 0000 gleichzeitig an das interne RAM und an das PROM gelegt. Die Ladeoperation ist beendet, wenn das letzte Wort vom externen PROM in das interne RAM eingelesen und gespeichert worden ist. Andere Operationen wie Program oder Verify haben den gleichen Ablauf der Adressierung. Die Beginnadresse ist die Adresse im RAM, zu der die Daten vom Adressenwort 0000 des externen PROMs eingespeichert werden.

Die Beginnadresse wird nach Drücken der Begin-RAM-Taste über das Tastenfeld eingegeben und durch die Enter-Taste festgelegt. Sie wird für alle Befehle die sich auf die externen PROMs beziehen oder wenn Daten über den seriellen Ausgang ausgegeben werden, benützt.

Durch die Festlegung der Anfangsadressen können Daten von unterschiedlichen PROMs nacheinander in das interne RAM eingeschrieben werden.

Beispiel: Der Inhalt von zwei kleineren PROM wird in das interne RAM geladen und auf diese Weise läßt sich ein größeres PROM programmieren. Der Inhalt von zwei 2708-EPROM wird in ein 2716-EPROM übernommen.

Bei der Dateneingabe über die Tastatur in das SYSTEM 19 wird verlangt, daß alle Datenübertragungen zwischen dem Baustein und dem RAM in den gesamten Adressenbereich mit eingeschlossen werden. Eine teilweise Programmierung, Laden oder Überprüfung ist nur mit der "Remote Control"-Wahlmöglichkeit durchführbar. Dies kann man auch erreichen, wenn die Daten in dem RAM entsprechend manipuliert werden.

Bei Eingangs- und Ausgangsfunktionen für einige Formate ist es notwendig, den Bereich der Programmieradressen über einen Datentransport umlaufen zu lassen und zu definieren. Die Begin-Adresse kennzeichnet die Startadressen im RAM, mit der Endadresse wird die maximale Adresse des Datentransfers festgelegt.

2.1.3 SUM-CHECK (Prüfsummenbildung)

Um eine einwandfreie Datenübertragung zu garantieren, berechnet das SYSTEM 19 die Prüfsumme. Diese besteht aus einer binären Summe aller übertragenen Daten, die binär addiert mit Übertrag 256 Modul und in hexadezimaler Form angezeigt werden. Der berechnete Wert, der nach jeder Datenübertragung vorhanden ist, muß immer bei den gleichgeladenen Daten dieselbe Summe ergeben. Dieser Wert wird in dem SYSTEM 19 abgespeichert und dient als vergleichender Wert für spätere Operationen. Bei nachfolgenden Kopien eines Masterprom werden somit sichergestellt, daß die Daten indentisch sind, wenn die angezeigte Prüfsumme (Sum-Check) nach dem Laden mit der früher notierten Prüfsumme identisch ist.

Die Prüfsummenbildung ist auch interessant für die Ein- und Ausgabeoperationen bei der Datenfernübertragung. Das gewählte Datenformat darf dabei keinen Fehler aufweisen. Die Bildung einer Prüfsumme zeigt an, ob die Datenübertragung fehlerfrei durchgeführt worden ist. Diese Prozedur schützt vor falschen Daten, die durch verstümmelte oder gelöschte Speicherelemente, Leitungsrauschen oder anderen Problemen der Datenfernübertragung auftreten können. Die Prüfsumme erscheint in der Adressenanzeige mit Dezimalpunkten.

2.1.4 Nachprüfung des internen RAM

Die vorausgegangene Beschreibung für die Prüfsumme sollte nicht mit der Nachprüfung des internen RAM verwechselt werden. Dies ist ein Wert der auf ähnliche Weise berechnet wird, wie die Prüfsumme, aber sich über das ganze RAM ausdehnt. Dieser Wert wird dazu benützt, um den gesamten RAM-Speicher zu überwachen. Wenn ein Wechsel in dem Inhalt des internen RAM durch einen externen Fehler auftritt, z.B. Speicherfehler oder Stromausfall, gibt das SYSTEM 19 sofort ein optisches und akustisches Signal ab.

2.1.5 Fehlererkennung

Das SYSTEM 19 behandelt Fehler auf eine bestimmte Art und Weise. Bei normalen Fehlern jedoch wird der Mikroprozessor des SYSTEM 19 das Problem auf eine der zwei Arten behandeln:

- 1) Der Programmierer gibt einen Dauerton ab, wenn eine verkehrte Taste gedrückt worden ist.
- 2) Bei einem fehlerhaften Programmierablauf folgt ein Ton. In der Anzeige erscheint "Err" und eine Nummer des Fehlers. Die Operation kann verbessert werden, nachdem die einzelnen Befehlsabschnitte berichtigt worden sind oder eine andere Operation versucht wird.

2.2 Betriebshinweise

Achtung!

Bitte lesen Sie sich die nachfolgenden Abschnitte aufmerksam durch.

2.2.1 Einschalten

Das SYSTEM 19 hat keine interne Netzspannung und muß daher an das allgemeine Stromnetz angeschlossen und auf den jeweiligen Spannungswert eingestellt werden. Ist keine Programmierkarte in dem Gerät, so sieht man in dem Ausschnitt der Platinenabdeckung am Gehäuseboden den Spannungswert.

Achtung!

Das SYSTEM 19 darf nur aus- oder eingeschaltet werden und der Programm PAK darf nur dann gewechselt werden, wenn sich kein PROM im Sokkeladapter befindet. Das SYSTEM 19 wird mit dem Schalter auf der Rückseite des Gerätes ein- oder ausgeschaltet. Beim Einschalten des Programmers erfolgt ein automatischer Ablauf eines Selbst-Testprogrammes. Wird dieser Test erfolgreich durchlaufen, dann leuchten die

Anzeigen TEST und COMPLETE auf. In dem Anzeigenfeld wird dann die maximale RAM Endadresse angezeigt. Beim Einschalten wird der Inhalt des RAM-Datenspeichers automatisch gelöscht.

2.2.2 Programmierkarte und Sockel

Vergewissern Sie sich immer vor der Anwendung des SYSTEM 19, ob die richtige Programmierkarte und der Sockel für das entsprechende Speicherelement (PROM) installiert sind. Beziehen Sie sich dabei unbedingt auf die DATA I/O Vergleichsliste "Comparison Chart", in welcher für die jeweiligen PROM die notwendigen Programmierkarten und Sockeladapter Nummern eingetragen sind. Nachdem alle Vorbereitungen getroffen worden sind, können Sie einschalten, wobei die Programmierkarte installiert sein soll. Ist das Gerät betriebsbereit, kann die Programmierkarte gewechselt werden, ohne das sich die Daten im RAM ändern. Dies gilt auch bei einem Wechsel des Speicherelementes. Bei dem Wechsel der Programmierkarte gibt das Gerät einen Dauerton ab. Ist eine neue Programmierkarte installiert, leuchtet die Lampe in der KEYBOARD-Taste auf.

2.2.3 PROM

Eine gute elektrische Verbindung zwischen dem PROM und dem Sockel auf der Programmierkarte ist unbedingt erforderlich. Setzen Sie das PROM in den Sockel, wenn der Hebel für den Spannverschluß nach oben gekippt ist. Versichern Sie sich unbedingt, daß der Pin 1 des PROM mit dem weißen Punkt auf dem Sockel übereinstimmt. Versichern Sie sich ebenfalls, daß die Pins sauber und nicht beschädigt (verbogen) sind. Mit dem Hebel wird das Speicherelement fest eingespannt.

Achtung!

Leuchtet das START-Licht auf, darf kein PROM aus den Sockel herausgezogen oder hineingesteckt werden.

2.3 Umgang mit PROMs

2.3.1 Betriebsart

Drücken Sie die PROGRAM-, LOAD- oder VERIFY-Taste. Ein Licht auf der betreffenden Taste leuchtet auf und zeigt die Funktion an. Durch Drücken der START-Taste wird die angewählte Funktion ausgeführt. Das Licht auf der START-Taste leuchtet während der Ausführung auf und zeigt dem Bediener, daß das SYSTEM 19 eine Operation ausführt. Alle Funktionen laufen automatisch ab.

Eine Operation ist abgeschlossen, wenn die COMPLETE-Anzeige aufleuchtet. Gleichzeitig erlischt die START-Leuchte.

Fehler während der Ausführung werden durch eine bestimmte Nummer in der Anzeige sichtbar. Es ergibt sich folgender Fehlercode:

SYSTEM 19 Fehleranzeige

Code	Name	Beschreibung
21	Illegal Bit Error	Für bereits programmierte Bits im Prom befinden sich Leerstellen (nicht gesetzte Bits) im RAM.
22	Programming Error	Die Programmierelektronik konnte das PROM nicht programmieren. (EnergieLimit)
23	First Pass Verify Error	Die Daten des PROMs waren falsch im ersten Durchlauf der automatischen Überprüfung nach der Programmierung des PROMs.
24	Second Pass Verify Error	Die Daten des PROMs waren falsch im zweiten Durchlauf der automatischen Überprüfung nach der Programmierung des Speicherelements.
25	No Programming Electronics Installed	Im Programm Pak ist kein Programmierkartensatz eingesteckt.
26	Programming Electronics Reset	Die Programmierelektronik kann nicht starten, evtl. Überstromabschaltung durch ein defektes PROM.
27	BEGIN RAM Too High	RAM-BEGIN Adresse zu hoch gewählt.
28	Calibrate Mode Address Range Error	Der Bediener versucht ein PROM bei einer falschen Adresse im Abgleichzustand zu programmieren.
41	Framing Error	Die serielle Schnittstelle erkennt das Startbit, aber das Stopbit wurde falsch gesetzt, evtl. falsche Baud-Rate.
43	Framing and Overrun Combined	Siehe 41 und 42
44	Parity Error	Parity Fehler für die eingegangenen Daten von der seriellen Schnittstelle.
46	I/O-Timeout	Keine Zeichen (oder nur Nullen und Rub-Outs) werden auf der seriellen Schnittstelle für 10 Sekunden nach dem Drücken der START-Taste angenommen.
48	Serial Port Buffer Overflow	Der serielle Eingangsspeicher bekommt zu vielen Zeichen.
51	END-RAM Below BEGIN-RAM	Die END-RAM-(Endadresse) oder die BEGIN-RAM-(Anfangsadresse) sind falsch eingegeben.
52	Input Compare Error	Die Daten der seriellen Schnittstelle stimmen beim Vergleich mit den Daten im RAM nicht überein.
53	No Data	Es werden keine ankommenden Daten erkannt.
54	Insufficient Data On Remote Input	Der Programmierer erwartet noch weitere Zeichen über die Schnittstelle.
55	Remote Input Sum-Check Error	Die Quersumme stimmt mit dem vorher berechneten und gespeicherten Wert der Quersumme nicht überein.
56	Address Range Error	Das Adressenfeld erhält Daten, die über dem Wert des adressierbaren Bereiches der internen RAMs liegen.

Code	Name	Beschreibung
57	Too Much Data	Obwohl die RAM-Endadresse erreicht ist, werden noch Daten eingegeben.
61	No RAM Installed	Es ist kein RAM-Datenspeicher vorhanden.
62	Last RAM Not On 1 k Boundary	Das RAM im Programmierer erreicht den von Wert von 1 k nicht, da ein Hardwarefehler vorliegt.
63	RAM Wire Failure	Der Programmierer konnte einen Wert der Daten nicht in das RAM einschreiben.
64	RAM Data Failure	Werte im RAM-Datenspeicher haben sich verändert.
66	Unidentified Interrupt	Die IRQ-Leitung zum Prozessor hat einen L-Pegel.

2.3.2 Setzen oder Überprüfen der Anfangsadresse

- a) Drücken Sie die BEGIN-RAM-Taste. Die Anzeige in der Taste leuchtet auf. Die momentane BEGIN-Adresse wird in dem Anzeigenfeld sichtbar. Die BEGIN-RAM-Adresse ist immer Null, wenn das SYSTEM 19 eingeschaltet wird.
- b) Wählen Sie eine neue Operation aus, wenn keine Veränderung benötigt wird oder geben Sie eine neue BEGIN-RAM-Adresse in das hexadezimale Tastenfeld ein.
- c) Wenn der eingegebene Wert korrekt war, drücken Sie die ENTER-Taste. Wenn dagegen ein Eingabefehler verursacht wurde, drücken Sie die BEGIN-RAM-Taste, um wieder den ursprünglichen Wert zu erhalten.
- d) Wenn die ENTER-Taste einen Ton auslöst, so ist der Wert der Anfangsadresse höher als die maximale RAM-Adresse. Drücken Sie die BEGIN-RAM-Taste, damit die fehlerhafte Eingabe rückgängig gemacht wird.

Vor der Datenanzeige leuchtet ein Dezimalpunkt auf, wenn Sie eine BEGIN-Adresse eingegeben haben. Eine Auswahl von anderen Funktionen (Progr., Load, Verify) ist hierbei nicht möglich, andernfalls wird ein Ton hörbar. Der Wert muß durch die ENTER- oder BEGIN-RAM-Taste zuerst bestätigt werden, bevor eine andere Funktion ausgewählt werden kann.

2.3.3 Setzen oder Überprüfen der Endadresse

Beim Ausgeben von Daten über die serielle Schnittstelle ist es möglich durch Eingeben einer Endadresse nur einen Teil der RAM-Daten auszugeben.

- a) Drücken Sie die SELECT-Taste
- b) Drücken Sie die E-Taste auf der hexadezimalen Tastatur, so daß die maximale Endadresse in der Adressenanzeige erscheint.
- c) Geben Sie die gewünschte RAM-Endadresse über die hexadezimale Tastatur ein.
- d) Wenn die eingegebene Adresse richtig war, drücken Sie ENTER. Wenn Sie einen Eingabefehler begangen haben, beginnen Sie wieder bei a).
- e) Wenn auf das Drücken der ENTER-Taste ein Ton folgt, so ist der END-RAM-Wert über der höchsten RAM-Adresse. Beginnen Sie wieder bei a).
- f) Nach der Festlegung der End-Adresse mit der ENTER-Taste bleibt die End-Adresse bis zum erneuten Eingeben oder dem Ausschalten des SYSTEM 19 gespeichert.

2.3.4 Laden eines Master PROM in das RAM

Beachten Sie hierzu Abschnitt 2.2.3 und fahren Sie wie folgt fort:

- a) Geben Sie eine Anfangsadresse ein, wenn nötig wie unter 2.3.2 beschrieben.
- b) Das Master Prom muß richtig im Sockel stecken.
- c) Drücken Sie die LOAD-Taste
- d) Drücken Sie die START-Taste

Während des Ladevorganges leuchtet die Anzeige in der START-Taste auf.

Bei einer BEGIN-RAM-Adresse 0000 wird das erste Datenwort des PROM in die erste Speicheradresse des RAM geladen. Die nachfolgenden Wörter werden in die fortlaufenden RAM-Adressen geladen.

Die Anzeige in der START-Taste erlischt, wenn alle Daten von dem Master PROM in das interne RAM übertragen worden sind und die Lampe "COMPLETE" aufleuchtet.

Die Adressenanzeige zeigt den Wert der Prüfsumme für das gerade geladene Master PROM an. Dies sollte man mit dem früher notierten Wert der Prüfsumme vergleichen. Die Prüfsumme ist von der üblichen Adressenanzeige durch die vier zusätzlichen Punkte in der Anzeige leicht zu unterscheiden.

Mit den meisten Programmierkarten wird die Übertragung vom PROM zum internen RAM mit der kleinsten Spannung durchgeführt. Dies schließt bei Pin-kompatiblen PROMs falsche Übertragungen aus, wenn mit einer verkehrten Programmierkarte eingelesen wird. Die VERIFY-Operation wird für die Überprüfung der maximalen Bedingungen verwendet. Setzen Sie aber nicht unbedingt voraus, daß ein Speicherelement (PROM) mit programmiertem Inhalt auch die gewünschten Daten enthält, es enthält die in das RAM eingegebenen Daten. Die Merkmale der Prüfsumme sollten dazu benutzt werden, die Datenreinheit oder Datenrichtigkeit zu garantieren.

2.3.5 Programmieren eines PROM

Sehen Sie bitte unter dem Abschnitt 2.3 nach und verfahren Sie wie folgt:

- a) Geben sie die Anfangsadresse ein, wenn erforderlich (Abschnitt 2.3.2).
- b) Das PROM muß richtig im Sockel stecken.
- c) Drücken Sie die PROG-Taste
- d) Drücken Sie die START-Taste.

Während des Programmiervorganges leuchtet die Anzeige in der START-Taste. Die Statusanzeigen können bei den automatischen Programmfolgen aufleuchten, wenn z.B. ein Fehler auftritt.

Das SYSTEM 19 zeigt jeden Fehler sofort an durch Blinken der Anzeige. Die Statusanzeigen während der automatischen Programmierung haben folgende Bedeutung:

TEST-Licht leuchtet:

Das SYSTEM 19 führt eine Leerprüfung und einen Illegal-Bit-Test des PROM im Sockel durch. Wenn zusätzlich die Anzeige aufleuchtet, so ist das PROM in verschiedenen Bit nicht leer. Die RAM-Adresse und die Daten in der ersten belegten Speicheradresse wird unmittelbar angezeigt. Durch Drücken der DEVICE-DATA-Taste veranlaßt die Anzeige die PROM-Adresse mit den Daten mitzuteilen. Wenn die PROM-Nichtleer-Anzeige übersprungen werden soll, drücken Sie die START-Taste um die nächsten Schritte der Programmierung einzuleiten.

PROGRAMM-Licht leuchtet:

Das PROM wird gerade mit den Daten aus dem internen RAM programmiert. Die Datenanzeige wechselt segmentweise, damit der Bediener erkennen kann, daß das SYSTEM 19 arbeitet. Dieser Schritt der Programmierung kann mehrere Minuten dauern.

VERIFY-Licht leuchtet:

Das SYSTEM 19 vergleicht jedes Datenwort des PROM mit dem entsprechenden Datenwort im internen RAM. Tritt ein Fehler auf, so wird dies durch ein Lichtsignal sichtbar. Ist der gesamte Speicherinhalt zwischen PROM und RAM überprüft worden, leuchtet das COMPLETE-Licht auf. Mehrere Tests werden während dieser Überprüfungsfolge durchgeführt, wobei sich für die jeweiligen Programmierungskartensets bestimmte Detailinformationen ergeben. Weitere Informationen, die einen Übertragungsfehler in dieser Operation erkennen, sind im Abschnitt 2.3.6 nachzusehen.

COMPLETE-Licht leuchtet:

Wenn keine Übertragungsfehler bei einer Operation aufgetreten sind, wird in der Adressenanzeige die Prüfsumme angezeigt. Mit der Datenanzeige wird der dezimale Wert der Summe der bereits programmierten PROMs angezeigt.

2.3.6 PROM Überprüfung

- a) Beachten Sie hierzu Abschnitt 2.2
- b) Setzen Sie die Anfangsadresse (siehe Abschnitt 2.3.2), wenn erforderlich.
- c) Das PROM muß richtig im Sockel stecken.
- d) Drücken Sie die LOAD-Taste
- e) Drücken Sie die START-Taste

Während des Vorganges leuchtet die Anzeige in der START-Taste auf.

Der Programmierer führt einen Vergleich zwischen den Daten des PROM und den RAM-Daten durch. Bei einer Nicht-Übereinstimmung wird die Anzeige aufleuchten und die RAM-Adresse mit Daten angezeigt. Die DEVICE-DATA-Taste veranlaßt den Programmierer, die Adresse des PROM mit Daten anzuzeigen. Nicht-Übereinstimmungen zwischen RAM und PROM können mit der START-Taste übersprungen werden.

Nachdem das PROM überprüft worden ist, erlischt das START-Licht und das COMPLETE-Licht leuchtet auf. Das Adressenfeld zeigt die Prüfsumme der Daten von dem PROM an. Die Prüfsumme wird aber nicht angezeigt, wenn ein Überprüfungsfehler aufgetreten ist.

Bemerkung:

Weil manche PROMs mit niedriger und erhöhter Versorgungsspannung überprüft werden, führt das SYSTEM 19 immer zwei Vergleichsdurchläufe aus.

2.3.7 DEVICE-DATA-Taste

Zeigt das SYSTEM 19 eine RAM-Adresse und die dazugehörigen Daten an, so können die entsprechenden PROM-Adressen und Daten überprüft werden.

Drücken Sie die DEVICE-DATA-Taste und die PROM-Adresse, die zu der angezeigten RAM-Adresse gehört, wird angezeigt. Die PROM-Adresse ist nur dann gleich mit der RAM-Adresse, wenn die BEGIN-Adresse gleich Null ist.

2.4 Dateneingabe über das Tastenfeld

Die Daten können von dem Tastenfeld im hexadezimalen Zahlensystem eingegeben werden. Dabei kann direkt auf eine Adresse gesprungen werden und von dieser kann dann schrittweise vorwärts oder rückwärts eine benachbarte Adresse angewählt werden. Die Daten an der angezeigten Adresse lassen sich wechseln, löschen oder es können neue Datenwerte eingegeben werden.

CHANGE: Die Daten in der angezeigten Adresse gehen verloren. Neue Daten von der Tastatur werden in dieser Adresse gespeichert. keine anderen Adressen sind von diesem Vorgang betroffen.

DELETE: Daten der angezeigten Adresse gehen verloren und die Daten von jeder höheren Adresse werden um eine Adresse verringert und füllen die freie Adressenstelle aus. In die höchste Adresse des RAM wird eine Null eingespeichert.

INSERT: Alle Daten von der angezeigten Adresse zur höchsten Speicheradresse werden um einen Platz nach oben verlegt. Die eingegebenen Daten werden in der angezeigten Adresse eingespeichert. Die Daten der höchsten RAM-Adresse gehen verloren.

2.4.1 Tastenfeldanwahl und Adressenauswahl

- a) Beachten Sie Abschnitt 2.3
- b) Drücken Sie die KEYBOARD-Taste. Die Anzeige in der Taste zeigt an, daß das Tastenfeld angewählt worden ist und in der Adressenanzeige erscheint der Wert der Adresse von dem internen RAM, die bei einer vorherigen KBD-Anwahl zuletzt angezeigt wurde.
- c) Ist die angezeigte Adresse die gewünschte Adresse, drücken Sie ENTER. Die Daten an dieser RAM-Adresse werden in dem Datenfeld angezeigt.
- d) Wird eine andere Adresse gewünscht, wählen Sie eine Adresse zwischen Null und der höchsten Adresse. Ist der Wert falsch eingegeben worden, drücken Sie die KEYBOARD-Taste und fangen neu an. Wenn der Wert richtig eingegeben ist, drücken Sie die ENTER-Taste. Ist der Adressenwert zu hoch, wird die ENTER-Taste einen Ton auslösen, ansonsten werden zusätzlich die Daten des RAM angezeigt.

2.4.2 Übertragung der Daten vom Tastenfeld zum internen RAM

- a) Wählen Sie eine Tastenfeldoperation bei der gewünschten Adresse aus (siehe Abschnitt 2.4.1)
- b) Um von der angezeigten Adresse zu einer höheren Adresse zu kommen, drücken Sie ENTER. Die angezeigte Adresse wird pro Tastendruck jeweils um eine Stelle erhöht.

- c) Um von einer angezeigten Adresse zu einer niederen Adresse zu kommen, drücken Sie die REVIEW-Taste. Die angezeigte Adresse wird pro Tastendruck um eine Stelle verringert.
- d) Wechseln Sie die Daten bei keiner Adresse, während Sie eine neue Adresse wählen. Der rechte Dezimalpunkt in dem Datenfeld wird aufleuchten, wenn Sie die Daten eingeben. Diese Daten werden von dem Programmierer entweder angenommen oder verworfen, wenn zu viele Male eine KBD-Taste gedrückt wurde.
- e) Um die Daten anzunehmen, drücken Sie ENTER. Die Daten werden angenommen und die Adresse wird um Eins erhöht oder Sie drücken REVIEW und die Daten werden ebenfalls angenommen, aber die Adresse verringert sich um Eins.
- f) Liegt ein Eingabefehler vor, bevor ENTER gedrückt wurde, drücken Sie die KEYBOARD-Taste. Das Datenfeld bleibt dunkel. Drücken Sie ENTER und die ursprünglichen Daten erscheinen wieder.

Alle anderen Tasten sind hierbei nicht zu verwenden, während der rechte Dezimalpunkt im Datenfeld leuchtet. Wird aber eine dieser Tasten trotzdem betätigt, werden Sie immer einen Ton von dem Programmierer hören. Damit wird der Bediener informiert, daß eine unzulässige Operation vorliegt.

2.4.3 EDIT-Funktion

Mit den angezeigten Adressen und Daten kann die EDIT-Taste gedrückt werden. Diese Taste aktiviert die ENTER- und die REVIEW-Taste. Die ENTER-Taste erhält die INSERT-Funktion und die REVIEW-Taste die DELETE-Funktion.

Um die zusätzlichen Daten in eine bestimmte Adresse einzuschreiben, wird zuerst die Adresse mit dem Tastenfeld eingegeben und dann die ENTER-Taste gedrückt. Es ergibt sich danach folgender Ablauf:

- a) Drücken Sie EDIT. Die Anzeige in dieser Taste leuchtet auf und bestätigt den EDIT-Betrieb.
- b) Geben Sie die Daten auf die angezeigte Adresse ein. Wenn Sie dabei einen Fehler machen, drücken Sie EDIT und beginnen wieder von vorne.
- c) Drücken Sie INSERT. Die neuen Daten werden auf die gewählte Adresse eingegeben und die nächsthöhere Adresse erscheint. Die ursprünglichen Daten auf dieser Adresse rücken in die nächsthöhere Adresse. Aus diesem Grund werden alle Daten die über dieser Adresse liegen um eine Adressenstelle nach oben verschoben. Die Daten in der wert höchsten Adresse gehen verloren.

Die Schritte "b" und "c" können beliebig oft wiederholt werden.

Um die Daten an einer bestimmten Adresse zu eliminieren, müssen Sie zunächst mit dem Tastenfeld die gewünschte Adresse eingeben und dann ENTER drücken. Danach ergibt sich folgender Ablauf:

- a) Drücken Sie EDIT. Die Anzeige in der Taste leuchtet auf und bestätigt den EDIT-Betrieb des SYSTEM 19.

- b) Drücken Sie REVIEW. Die Daten in dieser Adresse werden eliminiert. Die Daten von der nächsthöheren Adresse werden nach unten verschoben und angezeigt. Die anderen höheren Adressen verfahren mit ihren Daten ebenso. Die höchste RAM-Adresse wird mit Null geladen. Aufeinanderfolgende DELET-Tastenfolgen lösen weitere Dateneliminierungsvorgänge aus.

2.5 SELECT-CODE-Operationen

Die SELECT-Taste wird dazu benötigt, eine Reihe von Operationen auszuwählen. Alle diese Operationen erleichtern den Betrieb des SYSTEM 19 wesentlich. Die einzelnen Operationen können durch einen allgemeinen SELECT-Code aufgegriffen werden. Diese Codes für das SYSTEM 19 sind in der nachfolgenden Tabelle gezeigt:

A) Daten-Manipulations-Befehle

Code	Name	Beschreibung
A1	Nibble Swap	Wechselt höhere und niedrigere Befehlshälften von jedem 8-Bit-Datenwort in dem internen RAM
A2	Fill Memory with Variable	Füllt den Speicher von der angewählten Adresse an mit dem spezifizierten Datenwort nach Drücken der ENTER-Taste bis zur RAM-Endadresse.
A3	Complement Memory	Invertiert den gesamten Inhalt des RAM (Datenspeicher).
A4	Clear RAM	Löscht das gesamte RAM

B) Setz und Abfrage-Befehle

B0	Device Size	Zeigt den Adressenumfang und die Wortbreite des momentanen Aufrüstungsstandes an (bestimmt durch Programmiermodul und Sockeladapter)
B1	RAM Sum-Check	Zeigt die Quersumme über das gesamte RAM an.
B2	Configuration Number	Zeigt eine einzige Nummer an, mit der die Software-Konfiguration und Revisionsstand des SYSTEM 19 zu identifizieren ist.
B9	Display Test	Alle Anzeigenlampen leuchten auf.
C1	Calibrate ON	Wählt eine Abgleich-Software um die Angleichung einer Programmierkarte zu unterstützen.
C0	Calibrate OFF	Wechseln von der Calibrator-Routine zum normalen Betriebszustand
F4	Four Bit	Wählt 4-Bit-Datenwortbreite-Operation
F8	Eight Bit	setzt den Select-Code-Befehl F 4 außer Funktion. Es wird die vom Sockeladapter eingestellte Datenwortbreite (4- oder 8 Bit) wirksam.

Code	Name	Beschreibung
		verwendet. Standard wird eine Null dem Wagen-Rücklauf (CR) ausgegeben; irgendein Wert zwischen 00 und FF kann bestimmt werden für den Set-Null-Befehl.

C) Serielle Interface-Befehle

D1	Remote Input	Das SYSTEM 19 wartet auf die Eingabe von Daten in das interne RAM über dem seriellen Eingang.
D2	Remote Input (with Device Control)	Das SYSTEM 19 wartet auf die Eingabe von Daten in das interne RAM über dem seriellen Eingang. Der Programmierer gibt einen DC1-(X-ON)-Code aus, um das periphere Datenausgabegerät zu starten und gibt einen DC3-(X-OFF)-Code aus um die Übertragung zu beenden.
D3	Input Compare	Veranlaßt einen Vergleich von den ankommenden Daten am seriellen Eingang mit dem internen RAM.
D4	Input Compare (with Device Control)	Veranlaßt einen Vergleich von den ankommenden Daten am seriellen Eingang mit dem internen RAM. Der Programmierer gibt den DC1-(X-ON)-Code aus, um das periphere Datenausgabegerät zu starten und gibt einen DC3-(X-OFF)-Code aus, um die Übertragung zu beenden.
D5	Remote Output	Veranlaßt die Datenausgabe von dem Programmierer
D6	Data Output (with Device Control)	Der Programmierer gibt den Punch-ON-Code (DC2) aus, wenn die Datenübertragung starten soll und am Ende der Übertragung einen Punch-OFF-Code (DC4).

D) Manipulations-Befehle für Interface 990-1901

B3	Format Status	Der Programmierer zeigt am Display drei "zwei-digit-Zahlen" an:
----	---------------	---

ADRESS		DATA
50	01	10

Angewählte Schnittstellenformatierung
z.B. 50 = Hex ASCII
Space

Die Anzahl (Hexadezimal) der über die Schnittstelle nach dem Wagenrücklaufbefehl (CR) ausgegebenen Null-Charakter (Select-Code D9)

Die Anzahl (Hexadezimal) der über die Schnittstelle ausgegebenen Datenbyte pro Zeile (Select-Code D8)

Code	Name	Beschreibung
B4	Translator Error Status	Der Programmierer zeigt am Display drei "zwei-digit-Zahlen" an. Die Anzeige ist abhängig von der angewählten Schnittstellen-Formatierung.

ADRESS		DATA
04	00	00

Anzahl der fehlenden Daten-Charakter bei der Dateneingabe über die Schnittstelle der Formatierung 01-03, 30-32, 50-53 und 80-86

Anzahl der Sum-Check (Prüfsummen)-Fehler oder fehlende F-Zeichen beim Format 01, 02 und 03 bei Schnittstellenübertragung.

Anzahl der Parity Fehler

B5	Input Compare Error Count	Zeigt die Zahl (dezimal) der Fehler an, die festgestellt werden bei Vergleich der über die Schnittstelle ankommenden Daten mit dem RAM.
D7	Output Nulls	Es werden 50 Null-Charakter über die Schnittstelle ausgegeben.
D8	Select Record Size	Anwahl (hexadezimal) der gewünschten Zeichenzahl pro Zeile für Formate mit veränderbarer Zeichenzahl.

Die Auswahl-Codes sind in folgende drei Gruppen unterteilt:

- Setz- und Abfrage-Befehle. Diese werden sowohl für die Auswahl für die Datenübertragungsformate, also auch für das Setzen- oder Überprüfen von Parametern verwendet.
- Serielle Interface-Operationen und Befehle
- Datenmanipulationsbefehle

Drücken Sie die SELECT-Taste, damit das SYSTEM 19 einen SELECT-Code erkennen kann. Einige dieser SELECT-Codes erfordern weitere Parameterspezifikationen, wie sie die vorhin gezeigte Tabelle darstellt. Die folgenden Abschnitte geben die Tastenfolgen für jede Befehlsart an:

2.5.1 Setz- und Abfrage-Befehle

- Drücken Sie SELECT. In der Anzeige der Adressen erscheint "SEL" und die anderen Anzeigen sind dunkel..
- Geben Sie den SELECT-Code über das hexadezimale Tastenfeld ein.
- Drücken Sie ENTER oder START
 - ENTER veranlaßt die Daten in der angezeigten Adresse die verkehrten Werte eines benützten Parameters bei einer Operation

anzugeben. Geben Sie einen neuen Wert ein, wenn die Vorgabeanweisung nicht akzeptiert wird.

2) START führt die gewünschte Funktion aus.

2.5.2 Serielle Interface-Operationsbefehle

- a) Setzen Sie die Anfangsadresse, wie im Abschnitt 2.3.2 beschrieben.
- b) Setzen Sie die Endadresse, wie im Abschnitt 2.3.3 beschrieben.
- c) Drücken Sie SELECT.
- d) Wählen Sie die gewünschte Datenformatierung mittels Eingeben der entsprechenden Codezahl durch das Tastenfeld.
- e) Drücken Sie ENTER. Die Code Nummer wird angezeigt.
- f) Drücken Sie SELECT.
- g) Wählen Sie den gewünschten Schnittstellen-Steuerbefehl D1 bis D6 (Abschnitt 2.5) mittels Eingeben des entsprechenden Codewortes durch das Tastenfeld.
- h) Drücken Sie ENTER, dann Start. Wird nur Start gedrückt, so ist das Eingeben von Daten in ein vorgegebenes Adressenfeld nicht möglich.

3. Translator-Spezifikation

Das DATA I/O-SYSTEM 19 kann an alle peripheren Geräte mit serieller Schnittstelle angeschlossen werden, die mit dem ASCII-Code arbeiten. Drei Operationen sind möglich: Dateneingabe, Datenausgabe und Vergleichsfunktion zwischen Ein- und Ausgabe. In einer Eingabeoperation wird der Programmierer die Daten im ASCII-Hex-Code annehmen, übersetzen und im RAM speichern.

Die Richtigkeiten der empfangenen Daten darf dann vorausgesetzt werden, wenn eine Eingangs-Vergleichsfunktion durchgeführt worden ist. In dieser Operation werden die eingehenden Daten mit den gespeicherten RAM-Daten verglichen. Der Programmierer zeigt einen Fehler bei Datenungleichheit sofort an.

Mit den gespeicherten und überprüften Daten im RAM kann dann die PROM-Programmierung vorgenommen werden. Daten die im RAM gespeichert sind, lassen sich durch einen Befehl wieder über die serielle Schnittstelle ausgeben. Daten-Byte werden im entsprechenden Datenformat für Kopien, Speicherfunktionen oder für die Dokumentation ausgegeben. Die einzelnen Operationen sind durch die Eingabe eines "SELECT"-Codes über das Tastenfeld möglich. Es ergibt sich folgende Tabelle:

Format	SELECT-Code
Binary	10
DEC-Binary	11
ASCII-BNPF	05 (05)*
ASCII-BHLF	02 (06)*
ASCII-B10F	03 (07)*
5-Level-BNPF	08 (09)*
ASCII-Octal (Space)	30
ASCII-Octal (Percent)	31
ASCII-Octal (Apostrophe)	32
ASCII-Hex (Space)	50
ASCII-Hex (Percent)	51
ASCII-Hex (Apostrophe)	52
ASCII-Hex (Comma)	53
Intel-Non-Intellec	71
RCA-COSMAC	70
MOS-Technology	81
Motorola Exercisor	82
Intel Intellec 8/MDS	83
Signetics Absolute Object	85
Fairchild F8 Formulator	80
Tektronix Hexadecimal	86

Für die Übertragung von Daten ohne Startcodes sind diese (*) alternativen SELECT-Codes üblich.

3.1 Datenwörter

Die ASCII-Hex-Space (Format 50)-Daten können entweder im 4- oder 8-Bit-Format erscheinen. 8-Bit-Wörter benötigen zwei hexadezimale Zeichen, während ein 4-Bit-Wort nur ein hexadezimales Zeichen benötigt. Jedem 4- oder 8-Bit-Wort muß sofort ein ASCII-Space-Charakter (Leerzeile) folgen. Andere Kontrolleigenschaften, wie z.B. Zeilenvorschub und Wagenrücklauf können in den Datenstrom eingeschlossen werden, solange ein Datenwort direkt nach einer Leerzeile folgt.

3.2 Start und Ende

Der Start-Code im ASCII-Hex-(Space)-Format ist STX (Kontrolle B oder Hex 02) und der End-Code ist ETX (Kontrolle C oder Hex 03). Alle gültigen Daten werden zwischen dem Start- und End-Code übertragen.

3.3 Adressenfeld

Die einzelnen Daten werden nacheinander fortlaufend eingegeben. Die Eingabe der Blockanfangsadresse ist nicht erforderlich, wird aber angenommen. Nach jedem Datenwort steht ein Leerzeichen (bei Format 50) mit diesem wird vom Programmierer die Adresse weitergeschaltet. Die Blockadresse ist durch ein \$-Zeichen gekennzeichnet und dient dem leichteren Auffinden von Daten. Die Blockadresse wird hexadezimal angegeben und umfaßt beim Ausgeben jeweils einen Datenblock von 16 Wörtern x 8 Zeilen = 128 Wörter.

\$A 0000	\$A 0080	\$A 100	Adressenfeld
XX XX XX	XX XX XX	XX XX XX	Datenfeld
XX XX XX	XX XX XX	XX XX XX	
XX	XX XX		

Durch das Eingeben der Blockadresse ist es möglich nur eine oder mehrere Daten zu einer oder mehreren bestimmten Adressen einzugeben, z.B. zu der Hexadezimal-Adresse 100, 101, 102 sollen die Daten 1A 09 C3 geladen werden.

Eingabe:	STX	\$A100,	1A	09	C3	ETX			
	Start- zeichen	Anfangs- adresse	Daten	Leer- zeichen	Daten	Leer- zeichen	Daten	Leer- zeichen	End- zeichen

Carriage Return (LF) und Line Fee (LF) Befehle werden bei der Dateneingabe vom Programmierer nicht beachtet, beim Datenausgeben automatisch produziert.

3.4 Prüfsumme

Um ein Mittel für die Sicherheit bei einer Datenübertragung zu haben, muß ein Prüfsummenfeld direkt nach dem ETX folgen. Die Syntax für dieses Prüfsummenfeld wird in der nachfolgenden Darstellung gezeigt:

Ende der Daten	Prüfsummenfeld
.. .. HH HH HH	\$(End-Code)SHHHH,

Bemerkung: Das Prüfsummenfeld besteht aus vier Hex-Digits die zwischen "\$" und "," eingeschoben sind.
 Das Prüfsummenfeld folgt sofort nach dem End-Code.
 Das Prüfsummenfeld kann wahlweise auf die Eingabe folgen, wird aber beim Ausgeben der Daten vom SYSTEM 19 immer ausgegeben.

Eine Prüfsummenbildung ist eine 16-Bit-Addition von den Bit der übertragenen Datenwörter, die als 4-Hex-Digits angezeigt werden. Ein Übertrag vom 16ten Bit ist ausgeschlossen. Für diese Möglichkeit soll folgendes Beispiel gezeigt werden:

<u>Hex-Daten</u>	<u>Binär-Daten</u>
84	1000 0100
C1	1100 0001
62	0110 0010
24	0010 0100
01CB	0000 0001 1100 1011
Prüfsumme in Hexadezimal	Prüfsumme in binärer Form

Beziehen Sie sich auf Abschnitt 2.1.3 für den Gebrauch der Prüfsumme bei I/O-Operationen.

3.5 Eingangs-Charakterisierung

Die einkommenden Daten im ASCII-Space-Code-(50) werden von dem SYSTEM 19 entschlüsselt und gespeichert. Alle Daten müssen dann entsprechend der Hex-ASCII-Space Formatierung eingegeben werden. Wenn der Programmierer für eine 4-Bit-Operation eingestellt ist, können die Daten entweder in 4- oder 8-Bit-Wörtern eingegeben werden. In diesem Fall wird das Digit vor jeder Leerstelle als Datenwort interpretiert.

Wenn der Programmierer für 8-Bit-Operationen eingestellt ist, sind 4-Bit-Daten nicht erlaubt und der Programmierer nimmt nur die letzten zwei Digits vor jeder Leerstelle als Datenwort an.

Das ETX signalisiert das Ende der Datenübertragung und muß wenigstens von 16 beliebigen Zeichen gefolgt werden. "\$", wenn eine wahlweise Prüfsumme oder eine Adressenanwahl benützt wird. Sequentielle Blöcke von Daten können über die Schnittstelle eingegeben werden, wenn nach dem ETX des jeweiligen Datenblocks und den danach folgenden Zeichen ein STX eingegeben wird.

Bemerkung: Wenigstens 16 Charakter müssen von einem ETX folgen, um einen "time-out-error" bei einer Datenübertragung zu vermeiden.

Die Daten für eine Eingangs-Vergleichsoperation haben die gleichen Bedingungen, wie die für die Eingangsoperation.

3.6 Ausgangscharakterisierung

Die Adressengrenzen für eine Ausgabeoperation sind die Anfangs- und Endadresse. Diese können durch die in Abschnitt 2 beschriebenen Prozeduren entsprechend gesetzt werden.

Der Programmierer sendet alle Daten zwischen diesen beiden Produkten, aufgeteilt in Datenblöcke zu je 8 Zeilen. Eine Adresse wird für das erste Wort in jedem Block spezifiziert.

Nach dem ETX sendet der Programmierer die Prüfsumme der übertragenen Daten. Ein Beispiel von Ausgangsdaten wird in der nachfolgenden Abbildung gezeigt:

Nicht ausgedrucktes STX		
STX (Control B)	Anfangsadresse	
	\$A0000	
	HH HH HH HH HH HH HH HH HH HH HH HH HH HH HH HH	Jeder Datenblock besteht aus acht Zeilen mit jeweils 16 Zeichen
	HH HH HH HH HH.....HH HH HH	
	HH.....HH	
	HH.....HH	
	HH.....HH	
	HH.....HH	
	HH.....HH	
	HH.....HH	
	\$A0080	
	HH.....HH	Letztes Datenwort im RAM
	HH HH HH HH HH HH HH HH HH HH HH HH HH HH HH HH	
	HH HH HH HH HH HH HH HH HH HH HH HH HH HH HH HH	ETX\$\$HHHH
		Nicht ausgedrucktes ETX (Control C)
		Prüfsummenfeld

4. Befehle für die Datenübertragung

4.1 Vorbereitung

Für eine erfolgreiche Datenübertragung zwischen einem Sender und Empfänger müssen mehrere Faktoren beachtet und gewährleistet sein. Diese werden in den nachfolgenden Abschnitten erläutert.

4.1.1 Polarität

Bei Eingabe-Operationen von der Tastatur über ein Master-PROM oder über die Schnittstelle beziehen sich die eingegebenen Daten auf eine positive Logik und werden beim Programmieren unabhängig von dem verwendeten PROM-Typ auch auf positive Logik bezogen.

Beispiel:

PROM A
 Leerzustand
 00-Daten (Low-Pegel)
 in allen Adressen

PROM B
 Leerzustand
 FF-Daten (High-Pegel)
 in allen Adressen

Das Datenwort 1E = Binär 0001 1110

	A	
PROM	0000 0000	
programmiert	000X XXX0	

	B
	0000 0000
	XXX0 000X

Ergebnis in beiden PROMs steht nach der Programmierung das Datenwort 1E.

4.1.2 Kompatibilität von PARITY, STOP-Bit und BAUD-RATE

Für eine Datenübertragung zwischen einem entfernten Interfaceelement und dem Programmer müssen beide Geräte für PARITY, STOP-Bit und BAUD-RATE übereinstimmen. Einstellen der BAUD-RATE, STOP-Bit und PARITY siehe Bild 1.5 und 1.6 Seite 9b.

4.1.3 Wortbreite

Die Daten werden entweder in Worten von 4- oder 8-Bits ausgedrückt. Die Daten lassen sich in 8-Bit-Wörtern (Standard) eingeben, wenn keine 4-Bit-Datenübertragung spezifiziert worden ist.

4-Bit-Operationen können ausgeführt werden, wenn ein Sockeladapter für ein 4-Bit PROM installiert oder wenn eine 4-Bit-SELECT-Operation gewählt ist.

4.1.4 Leer- und Startbedingung

Die Startbedingung einer Datenübertragung kann von beliebiger Länge sein und dürfen alle Charakteristiken, außer dem Startcode des Formats enthalten. Ein Lochstreifen, der in den Programmer eingegeben werden soll, darf jede Nummer der Leerstellencharakteristik am Anfang und Ende enthalten. Wenn die Daten zu einem Lochstreifenstanzer ausgegeben werden, wird der Lochstreifen eine Startbedingung von 50 Leerstellen, dann die Daten im spezifischen Format und anschließend weitere 50 Leerstellen enthalten.

Der Programmer muß genügend Leerstellen nach einem Funktionscode für den Wagenrücklauf senden, um gewisse Terminals an einem Leerlauf zu hindern. Setzen Sie die Leerstellen durch einen SELECT-Code-Befehl und sie geben dem Programmer eine entsprechende Pausenkapazität. Das Terminal enthält damit genügend Leerstellen, um sich nach jedem Wagenrücklauf die Daten zu erhalten. Der nächste SELECT-Code und die Ausführung für die Leerstellen sind im Abschnitt 1 nachzulesen.

4.1.5 Start- und Endcode

Der ASCII-Hex-(Space)-Datentransport beginnt mit einem Start-Codewort, einem nicht ausgedruckten STX (bzw. Control B). Bei der Ausgabe wird der Start-Code sofort nach den 50 Leerstellen der Startbedingung gesendet.

Eingabeoperationen werden mit einem End-Code, einem nicht ausgedruckten ETX oder wenn die Eingabe nicht unterbrochen wird durch das Ende des RAM oder die Keyboard-Taste ausgeführt. Eine neue Eingabeoperation kann durch einen STX innerhalb von 16 Charakteristiken nach dem ETX erfolgen.

Anhang

Programmierbare Halbleiter-Speicher

In der Elektronik, Datenverarbeitung und Mikroprozessortechnik unterscheidet man prinzipiell zwischen zwei Speichertypen, den RAMs (Random Access Memory = Schreib-Lese-Speicher) und den ROMs (Read Only Memory = Festwertspeicher).

Die Schreib-Lese-Speicher finden allgemein als Zwischen- und Datenspeicher ihre Anwendung. In ein RAM können Daten eingeschrieben (write) und zu einem späteren Zeitpunkt wieder zerstörungsfrei ausgelesen (read) werden. Schaltet man die Versorgungsspannung eines RAM aus, geht der gespeicherte Dateninhalt unweigerlich verloren.

Festwertspeicher finden allgemein als Befehls- und Steuerspeicher ihre Anwendung. Der Speicherinhalt (Programm) wird durch einen Anwender festgelegt und der Baustein entsprechend programmiert. Der Inhalt wird zerstörungsfrei ausgelesen (read) und bleibt auch dann erhalten, wenn die Versorgungsspannung ausgeschaltet wird.

Die Bezeichnung "ROM" für NUR-LESE-Speicher wurde um 1950 in der Datenverarbeitung geprägt. Die ersten Festwertspeicher wurden als Masken-ROMs bekannt. Für jedes Programm mußte eine separate Maske hergestellt werden und danach wurde der Baustein produziert. Programmänderungen konnten nur durch eine Maskenänderung vorgenommen werden. Der Anwender mußte Wochen auf seine Festwertspeicher warten.

Aus der Weiterentwicklung der Technologie entstanden die PROMs (programmierbare Festwertspeicher). Durch den Einsatz eines Programmiergerätes kann sich nun der Anwender seine Bausteine selbst herstellen und sofort sein Programm testen. Die Entwicklung brachte neben zahlreichen Technologien auch die EPROMs (löschbare Festwertspeicher). Nach der Entwicklung eines Gerätes wird der gespeicherte Inhalt eines EPROM durch UV-Licht gelöscht und das EPROM steht für neue Entwicklungen wieder bereit.

Voraussetzungen für ein optimales Arbeiten im Labor ist ein universelles Programmiergerät für möglichst viele Technologien und Speicherorganisationen.

1. Programmiertechnologien

1.1 Festwertspeicher nach dem Fusible-Link-Verfahren

Festwertspeicher nach diesem Verfahren können nach ihrer Programmierung nicht mehr gelöscht werden. Ein Nachprogrammieren ist möglich, wenn der betreffende Speicherplatz noch unprogrammiert ist.

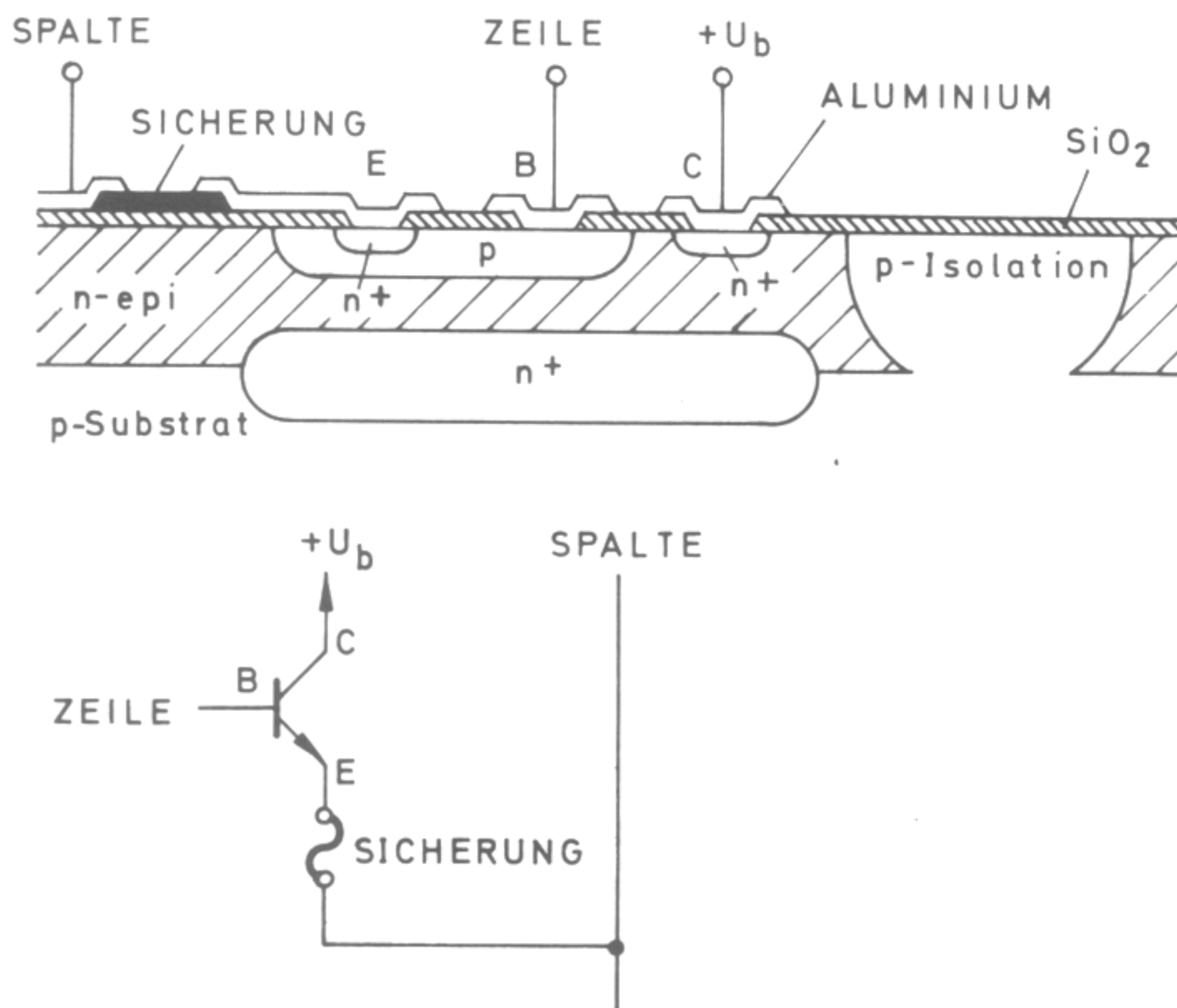


Bild 1.1 Festwertspeicher nach dem Fusible-Link-Verfahren

Bild 1.1 zeigt den Aufbau einer bipolaren Speicherzelle mit einem Sicherungselement, wie dies bei den meisten FL-PROMs üblich ist. Das Sicherungselement (fuse) liegt zwischen dem Emitter des Selektions-Transistors und der Spaltenleitung (COLUMN). Erhält der Transistor durch die Zeilenleitung (ROW) einen H-Pegel, also ein Selektionssignal, schaltet der Transistor durch. Ist die Zelle unprogrammiert, kann ein Strom über den Transistor und dem intakten Sicherungselement in die Spaltenleitung fließen und der Ausgangsverstärker erkennt einen H-Pegel an. Der H-Pegel kann durch einen invertierenden Ausgangsverstärker noch negiert werden. Ist die Zelle aber programmiert, d.h. das Sicherungselement ist unterbrochen, erkennt der Ausgangsverstärker einen L-Pegel an.

Das Sicherungselement ist je nach Technologie eine dünne Widerstandsbrücke, die durch einen genau definierten Stromstoß (Programmierimpuls) durchgeschmolzen wird. Dadurch ist keine Reprogrammierung (Löschung) möglich.

Wird die vom PROM-Hersteller vorgeschriebene Programmierspezifikation nicht exakt eingehalten, kann bei einigen PROM-Typen ein Zurückwachsen (grow back) der Widerstandsbrücken erfolgen, da bei der Programmierung metallurgische Rückstände vorhanden sind. Die Programmierspezifikationen, insbesondere über Leistung und Impulslänge des Programmierimpulses sind so ausgelegt, daß ein breiter Spalt zwischen Emitter und der Spaltenleitung entsteht. Bei richtiger Programmierung haben FL-PROMs eine praktisch unbegrenzte Lebensdauer.

1.2 Festwertspeicher nach AIM-Verfahren

Die Bezeichnung "AIM" steht für "Avalanche Induced Migration". Im Gegensatz zu den FL-PROMs sind AIM-PROMs reine Halbleiterspeicher.

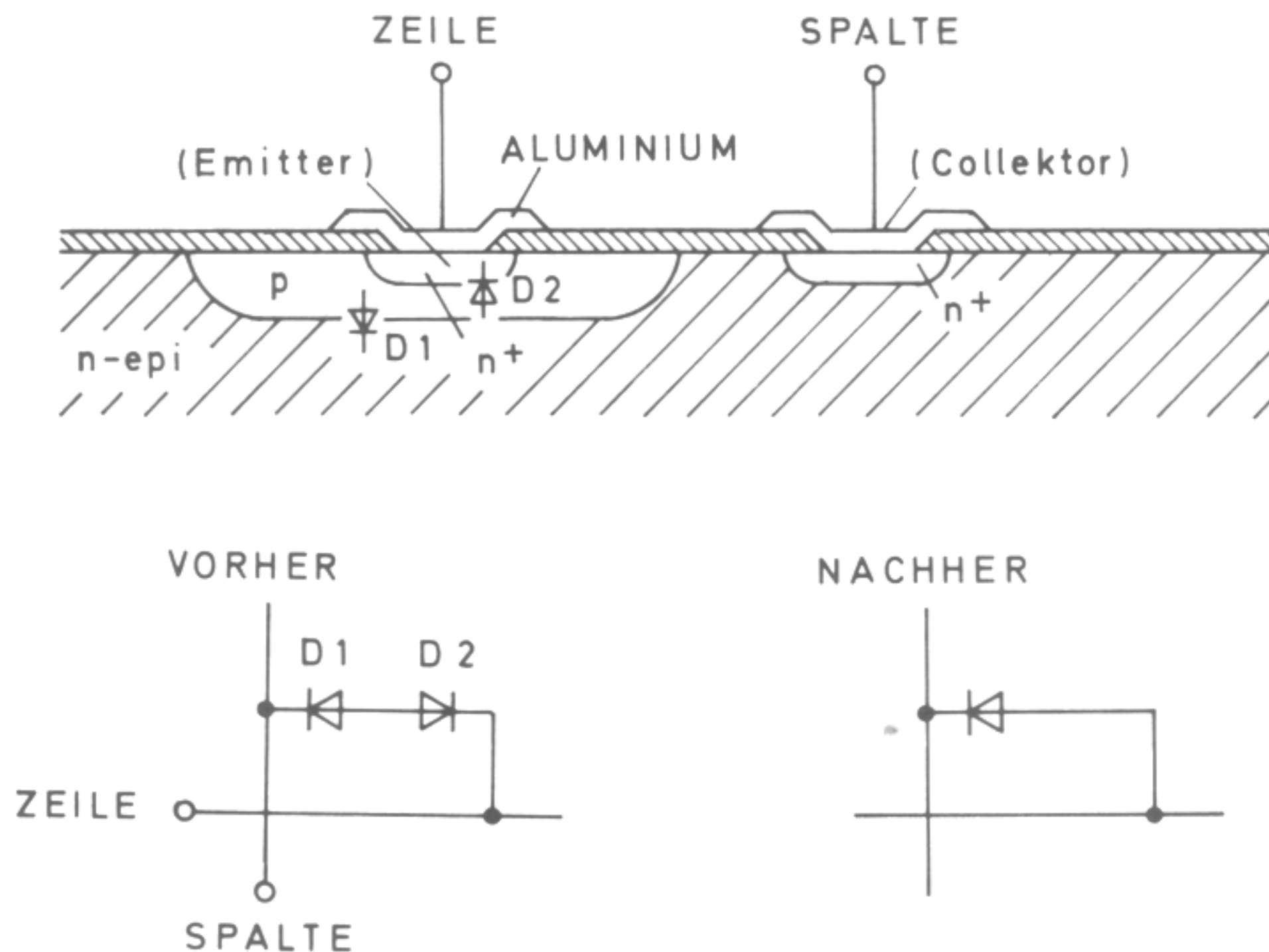


Bild 1.2 Festwertspeicher nach AIM-Verfahren

Bild 1.2 zeigt den Aufbau einer Speicherzelle für AIM-PROMs. Vor dem Programmieren sind zwei Dioden vorhanden, d.h. die beiden Dioden stellen einen Transistor dar. Der Kollektoranschluß des Transistors ist mit der Spaltenleitung und der Emitter mit der Zeilenleitung verbunden. Der Basisanschluß wird nicht benötigt und ist daher nicht angeschlossen.

Beim Programmieren wird die Basis-Emitter-Diode D2 der zu programmierenden Speicherzelle durch definierte Stromimpulse abgebaut und in einen Widerstand umgewandelt. Die Basis-Emitter-Diode hat im unprogrammierten Zustand einen Übergangswiderstand in der Größenordnung von 100 k Ω . Durch die Programmierimpulse reduziert sich der Übergangswiderstand auf 200 Ω . Im unprogrammierten Zustand kann durch die Basis-Emitter-Diode kein Strom fließen, da die Diode in Sperrichtung arbeitet.

Beim Programmieren einer AIM-Speicherzelle wandern Ionen der Aluminiumschicht in das n-Material des Emitters ab. Die Aluminium-Ionen besetzen die Störstellen in dem n-Material und es entsteht ein ohmscher Widerstand. Die Materialwanderung ist abgeschlossen, wenn genügend Aluminium-Dioden die Störstellen besetzt haben. Nach jedem Programmierimpuls muß der Übergangswiderstand der Diode gemessen werden, bis der definierte Übergangswiderstand erreicht worden ist.

Durch diese Reihenfolge, programmieren-messen-programmieren, kann die Materialwanderung laufend überprüft werden.

Werden beim Programmieren die Spezifikationen überschritten, kommt es in der betreffenden Diode zu einem Wärmestau und die Speicherzelle wird zerstört. Unterschreitet man die Spezifikation, erfolgt keine Materialwanderung, bzw. kein Abbau der Diode. Bei richtiger Programmierung haben AIM-PROMs eine unbegrenzte Lebensdauer.

1.3 UV-Löschbare Festwertspeicher

Bei den EPROM (Erasable PROM) oder REEPROM (RE-Programmable ROM) kann mit Hilfe von ultraviolettem Licht der gesamte Speicherinhalt gelöscht werden. Das Löschen von einzelnen Speicherdaten ist aber nicht möglich, es wird immer der gesamte Inhalt gelöscht. Nach einem Löschvorgang lassen sich neue Daten einschreiben. Bei genauer Einhaltung der Programmierspezifikation sind einige 100 Lösch-Programmierzyklen möglich.

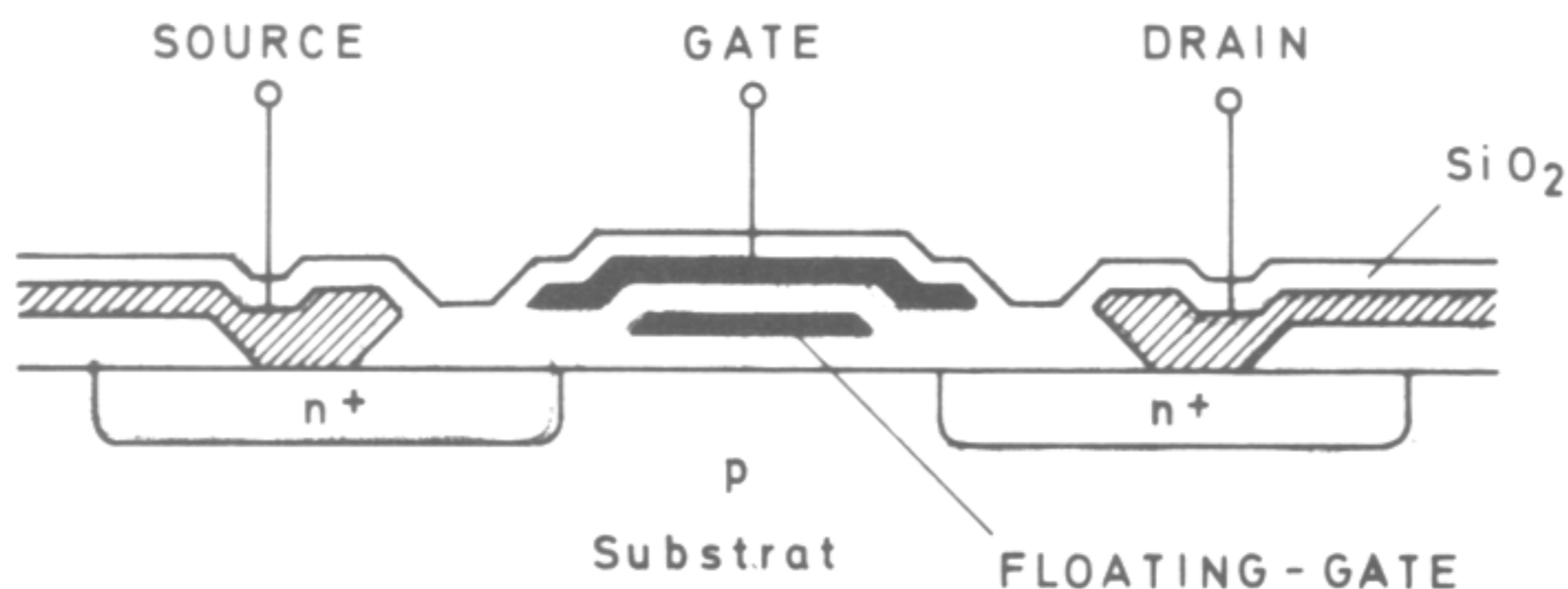


Bild 1.3 UV-Löschbarer Festwertspeicher

Bild 1.3 zeigt den Aufbau einer Speicherzelle für EPROMs. Die Speicherzelle besteht aus einem einzigen MOS-Transistor, der mit zwei Poly-Siliziumschichten versehen ist. Die obere Siliziumschicht ist der Select-Gateanschluss und dieser ist mit der Zeilenleitung verbunden. Die zweite Schicht ist ein offenes Gate (floating gate) und dient zur Informationsspeicherung. Beim Programmieren einer Speicherzelle wird durch eine Injektion energiereicher Elektronen die Speicherschicht aktiviert. Diese Ladung wird gespeichert, da keine elektrische Verbindung zu den umliegenden Schichten besteht. Durch die gespeicherte Ladung ändert sich die Schwellwertspannung der Zelle entsprechend.

Im unprogrammierten Zustand ist bei den meisten PROMs ein H-Pegel gespeichert, d.h. es ergibt sich eine sehr niedrige Schwellwertspannung durch die Verwendung der Silizium-Gate-Technologie. Bei der Ansteuerung des Select-Gates kann der Transistor einer Speicherzelle durchschalten zwischen Source und Drain. Der Ausgangsverstärker gibt einen H-Pegel ab. Im programmierten Zustand wird die Schwellwertspannung wesentlich angehoben. Bei einer Ansteuerung des Select-Gates kann der Transistor nicht durchschalten und der Ausgangsverstärker gibt einen L-Pegel ab.

EPROMs werden mit einem sehr starken UV-Licht der Wellenlänge von 2537 Å gelöscht.

1.4 Elektrisch löschbare Festwertspeicher EAROMs

EAROMs (Electrically Alterable ROM) sind Bit für Bit programmierbar und typenunterschiedlich durch hohe Spannungsimpulse als Ganzes oder wortweise löscherbar.

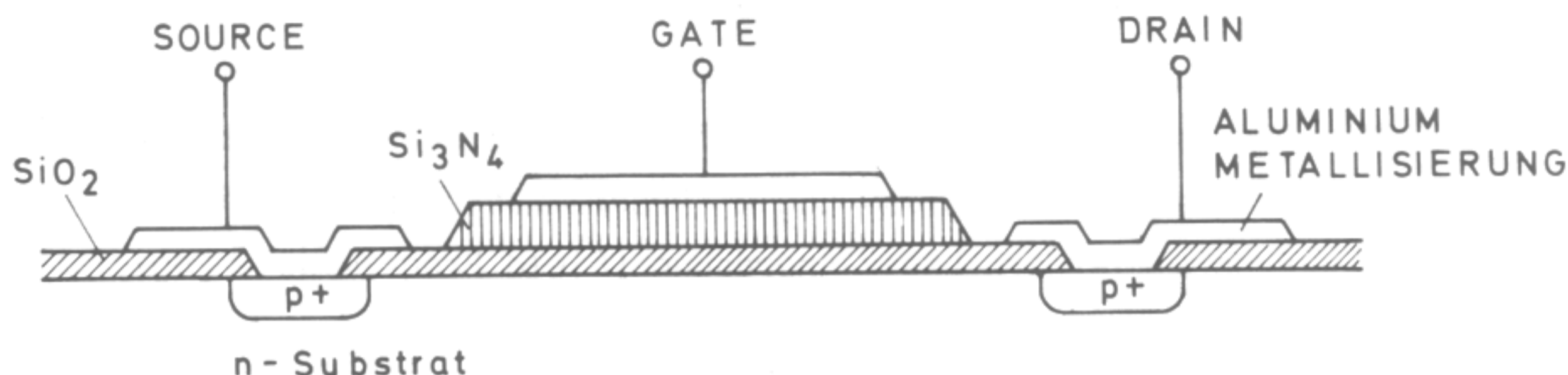


Bild 1.4 MNOS-Speicherzellenstruktur

Die Speicherzelle von Bild 1.4 besteht aus einem MNOS-Feldeffekttransistor (Metal Nitride Oxide Semiconductor). Zwischen dem Gateanschluß und der SiO₂-Schicht befindet sich das bewegliche Gate (floating gate) zur Speicherung der Information. Dabei handelt es sich um eine Zwei-Lagen-Isolation, die aus Siliziumnitrid und Siliziumoxid besteht.

Durch Anlegen einer positiven oder negativen Spannung läßt sich das Gate nach oben oder unten verschieben. Dadurch ändert sich auch die Wirkung der Gatespannung auf das n-Substrat. Bei einer positiven Spannung an dem Gate werden Elektronen aus dem n-Substrat in das Siliziumnitrid geschoben und daher ändert sich auch die Schwellerspannung des MNOS-Transistors. Bei einer negativen Spannung werden dagegen die Elektronen aus dem Siliziumnitrid in das n-Substrat zurückgeschoben. Durch die Verschiebung der Elektronen ergibt sich entweder ein Kanal zwischen beiden p-Schichten (Source und Drain) oder der Kanal wird abgebaut.

Legt man das Gate auf Masse, kann der MNOS-Transistor die Information speichern, selbst wenn die Versorgungsspannung abgeschaltet wird. Die Speicherung der Information bleibt über lange Zeit festgehalten.

Das Löschen der Informationen erfolgt durch Anlegen eines positiven Spannungsimpulses, dessen Impulshöhe und Impulsbreite genau definiert sein muß. Durch den Löschevorgang werden die Schwellerspannungen auf H-Pegel gebracht.

Für breitere Anwendungen haben sich die EAPROMs noch nicht einführen lassen, da die Zahl der Schreib-Lösch-Zyklen begrenzt ist auf etwa $10^6 \dots 10^8$. Mit zunehmender Zahl von Schreib-Lösch-Zyklen wird die Speicherzeit immer kürzer. An einer Verbesserung wird in einer Reihe von Firmen gearbeitet.

2. Programmierbare Logik-Bausteine

Die Halbleiterindustrie bietet heute zahlreiche programmierbare Bausteine an, die sich nicht nur durch die Technologie, sondern im Aufbau der Speicherorganisation (Terminologie) wesentlich unterscheiden. Die wichtigsten Speicherterminologien sollen kurz gezeigt und erläutert werden:

PROM : Festwertspeicher mit Adressendecoder oder (Programmable Read-Only-Memory).

DM : Programmierbare Dioden-Matrix (Diode Matrices)

PMUX : Programmierbare Multiplexer (Programmable Multiplexer)

FPGA : Programmierbare Gatter-Anordnungen (Field Programmable Gate)

PLA : Programmierbare Array-Logik (Programmable Array Logic)

FPLA : Programmierbare Logik-Anordnung (Field Programmable Logic Array)

Der optimale Einsatz dieser vielen Speichertypen einschließlich der Technologien bringt nicht nur für den Anwender große Probleme, sondern stellt auch an das Programmiergerät hohe Anforderungen. Universelle Programmiergeräte wie das SYSTEM 19 von Data I/O sind in der Lage mittels auswechselbarer Personality Modulen diese Programmierungen durchzuführen.

2.1 Programmierbare Festwertspeicher (PROM-Programmable Read-Only-Memory)

Die "Programmable Read Only Memories" oder PROMs sind die am meisten verwendeten programmierbaren Festwertspeicher. Über die Adressen von A_0 bis A_9 erfolgt die Adressierung der 8192 Speicherplätze von Bild 2.1.

Diese Adressierung erfolgt über den X-Decoder für die Zeilen, insgesamt sind 64 Zeilen in der Speichermatrix und über den Y-Decoder für die Spalten, insgesamt sind 16 Spalten vorhanden. Auf diese Weise wird jeder Speicherplatz in der Matrix erreicht und man spricht von einem voll adressierbaren PROM.

Das Auslesen einer Zeile erfolgt über die acht Datenausgänge O_0 bis O_7 . In Verbindung mit den Adressen ergibt sich folgende Terminologie zur Berechnung:

Konfiguration = $N \cdot M$

N = Anzahl der Adressen

M = Anzahl der Ausgänge

Für das PROM von Bild 2.1 gilt:

$1024 \cdot 8 = 8192$ Speicherplätze

Durch die 10 Eingänge lassen sich $2^{10} = 1024$ Adressen ansteuern. Jede dieser Adressen beinhaltet ein Speicherwort mit 8 Bit.

Bei der Programmierung der einzelnen PROM Typen muß auf die Wertigkeiten der Ausgänge geachtet werden, ob die Ausgänge einen aktiven H- Oder L-Pegel aufweisen können.

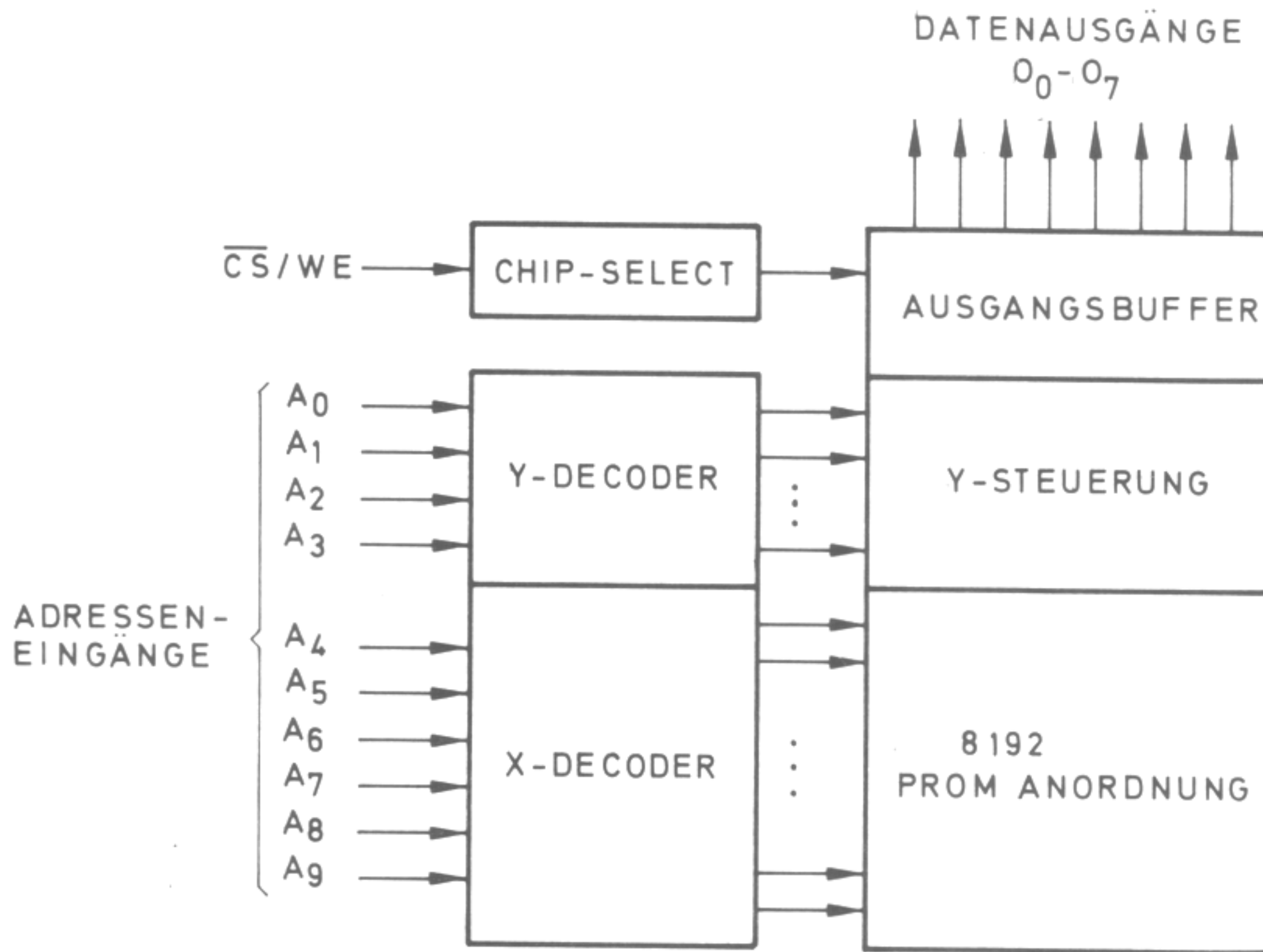


Bild 2.1 Aufbau eines 8192 Bit PROM

Dementsprechend soll die Programmieretabelle ausgefüllt werden. Mit dem CS-WE-Eingang erfolgt die Steuerung des Ausgangsbuffers. Hierbei unterscheidet man zwischen dem "offenen Kollektor" und "Tri-State"-Ausgang. Mit einem "offenen Kollektor" läßt sich durch eine "wired-AND"-Verknüpfung einfach eine Speichererweiterung realisieren. Dagegen kann ein Ausgang mit Tri-State-Verhalten direkt an jedem Datenbus angeschlossen werden.

Bipolare PROMs sind entweder in FL- oder AIM-Technologie aufgebaut und haben in der Regel folgende Speicherorganisationen: 32 x 8, 256 x 4, 256 x 8, 512 x 4, 512 x 8, 1024 x 4, 1024 x 8. Die typische Zugriffszeit (access time) liegt zwischen 40 ns und 80 ns. Es ergeben sich sehr schnelle Zugriffszeiten. Verwendet man ECL-PROMs in FL-Technologie, verringert sich die Zugriffszeit auf 15 ns bis 20 ns.

Hierzu stehen die relativ langsamen EPROMs mit einer Zugriffszeit zwischen 450 ns und 1000 ns. Die Organisation für diese EPROMs sind: 256 x 8 (2k), 1024 x 8 (8k), 2048 x 8 (16k), 4096 x 8 (32k) und 8192 x 8 (64k). Parallel zu diesen EPROMs sind maskenprogrammierte Festwertspeicher mit der gleichen Pinbelegung und gleicher Betriebsspannung erhältlich. Bei einer spezifizierten EPROM-Programmierung können diese direkt als Vorlage zur Erstellung der Masken-ROMs verwendet werden. Dadurch lassen sich die Herstellungskosten deutlich reduzieren.

Die "Electrically Erasable PROMs" kurz EAPROMs sind ähnlich wie die EPROMs aufgebaut. Sie werden mit den Organisationen 32 x 16, 256 x 4 (1k), 1024 x 4 (4k) und 2048 x 4 (8k) geliefert. Der Speicherzugriff liegt aber zwischen 950 ns bis 10 ms beim Auslesen der Daten, die Schreibzeit für ein Wort dauert 1 ms bis 50 ms und die Löschzeit pro Wort zwischen 19 ms und 100 ms.

2.2 Programmierbare Dioden-Matrix (DM Diode Matrices)

Die programmierbare Dioden-Matrix wird eingesetzt bei einfachen Logik-anwendungen z.B. als Decoder, Codierer, Umcodierer. Innerhalb dieses Bausteines sind 25 bis 48 Dioden in einer Matrix angeordnet. Zu jeder Diode gehört ein Sicherungselement für die Programmierung.

Die Ansteuerung der einzelnen Speicherzellen erfolgt über die Zeilen-leitung. Die Spaltenleitung ist mit dem Ausgangsverstärker verbunden.

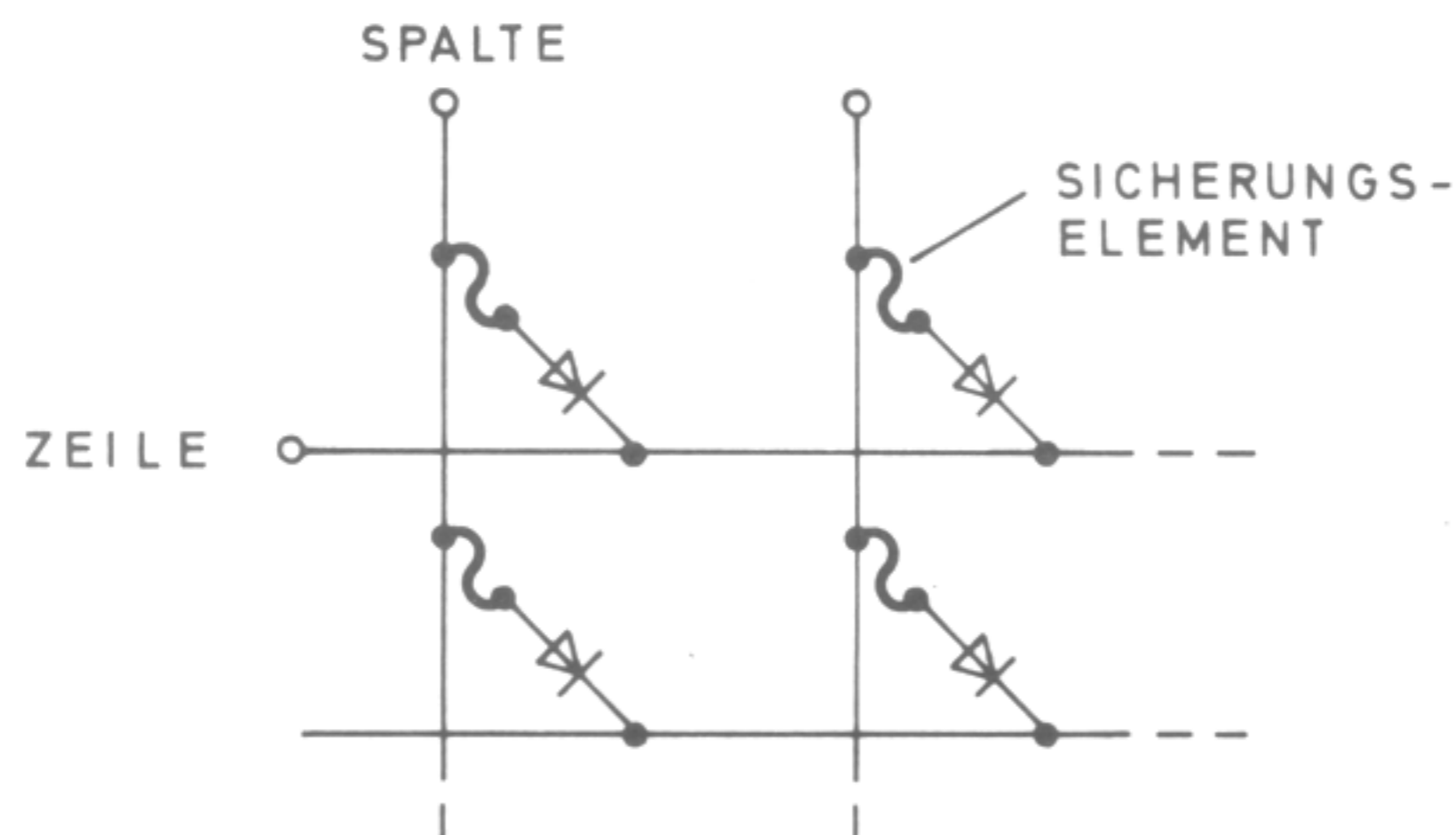


Bild 2.2 Programmierbare Dioden-Matrix

Bild 2.2 zeigt den Aufbau einer programmierbaren Dioden-Matrix. Liegt an der Zeilenleitung ein L-Pegel, kann über die Dioden ein Strom abfließen, da die Dioden in Durchlaßrichtung geschaltet sind. Die Spaltenleitung hat durch den Stromfluß ebenfalls einen L-Pegel, wenn das Sicherungselement nicht programmiert worden ist. An dem Ausgang liegt ein L-Pegel an. Bei einem H-Pegel an der Zeilenleitung kann kein Strom fließen und der Ausgang hat einen H-Pegel.

Ist das Sicherungselement durch einen Programmierimpuls abgeschmolzen, kann bei einem L-Pegel an der Zeilenleitung kein Strom abfließen und am Ausgang erscheint ein H-Pegel.

Mit dem Sicherungselement bestimmt man, ob zwischen der Zeilen- und Spaltenleitung eine elektrische Verbindung vorhanden ist. Bei einer elektrischen Verbindung erfolgt keine Programmierung, bei keiner Verbindung wird das Sicherungselement abgeschmolzen.

2.3 Programmierbare Multiplexer (PMUX Programmable Multiplexer)

Die programmierbaren Multiplexer oder PMUX-Speicher sind eine relativ neue programmierbare Speicherlogik.

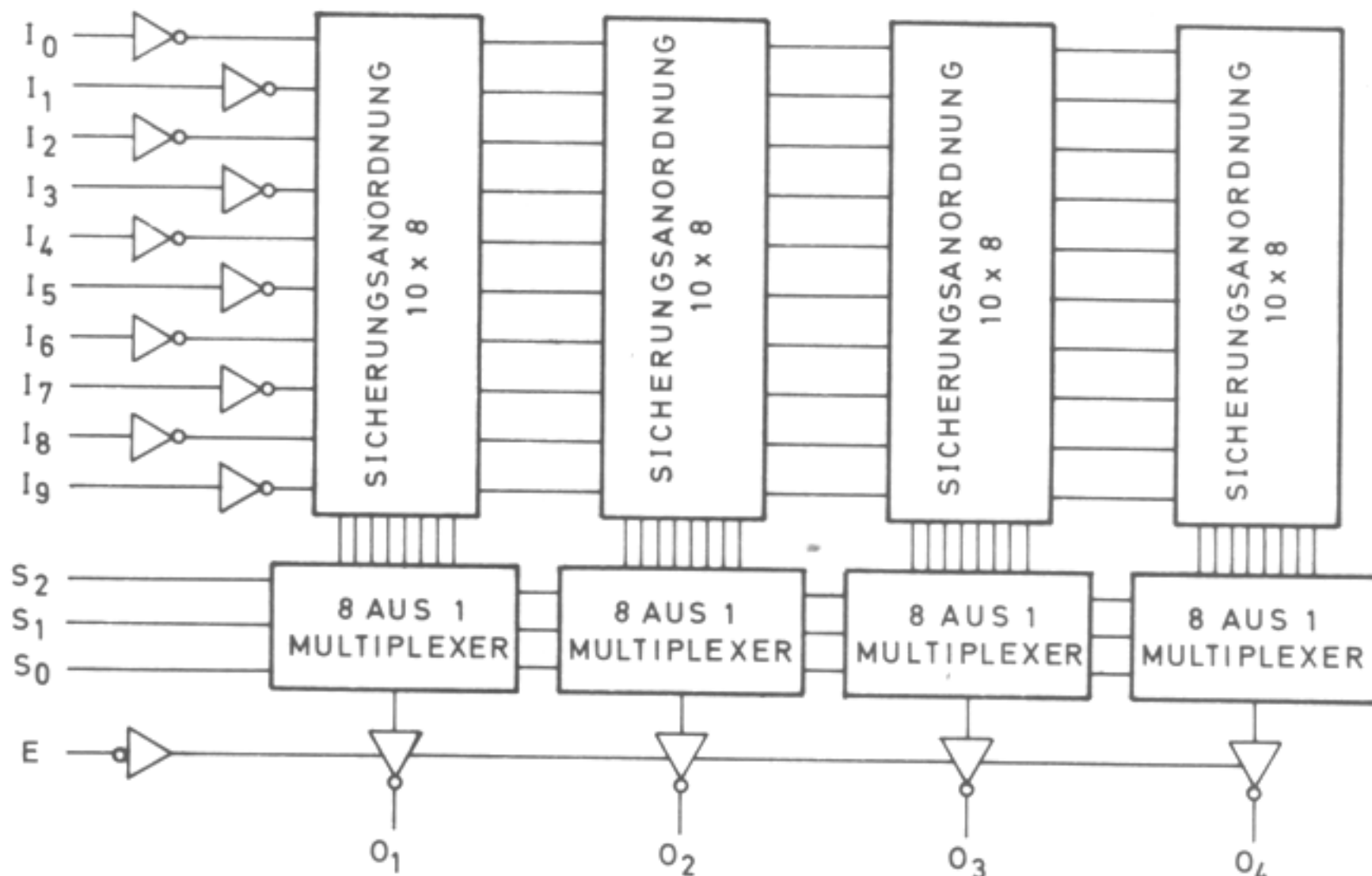


Bild 2.3 Blockdiagramm eines 10 x 8 x 4 - PMUX

Bild 2.3 zeigt den Aufbau eines 10 x 8 x 4-PMUX-Speichers. Mit den Eingängen I_0 bis I_9 bestimmt man die Zeilen innerhalb der vier Speicheranordnungen. Die zehn Zeilen liegen im Gegensatz zu den PROMs direkt an dem Speicher. Jede Zeile innerhalb einer Anordnung steuert acht Spaltenleitungen an. Die Verbindung zwischen Zeile und Spalte erfolgt über ein Sicherungselement mit Entkopplungsdiode.

Mit den drei Steuereingängen S_0 bis S_2 bestimmt man, welche Spaltenleitung auf den Ausgang geschaltet werden soll. Die 8 zu 1 Multiplexer wählen aus einem binären Steuerwort die betreffende Spalte aus und verbinden sie mit den Ausgängen von O_1 bis O_4 . Die drei Steuereingänge ergeben acht Kombinationen, denn $2^3 = 8$.

2.4 Programmierbare Gatter-Anordnung (FPGA Field Programmable Gate Array)

Unter dem Begriff "Field Programmable Gate Array) kurz FPGA versteht man eine einfache Anordnung von mehreren NAND-Gattern mit nachgeschalteten Exklusiv-ODER-Gattern. Bild 2.4 zeigt den Aufbau dieses Speichertyps.

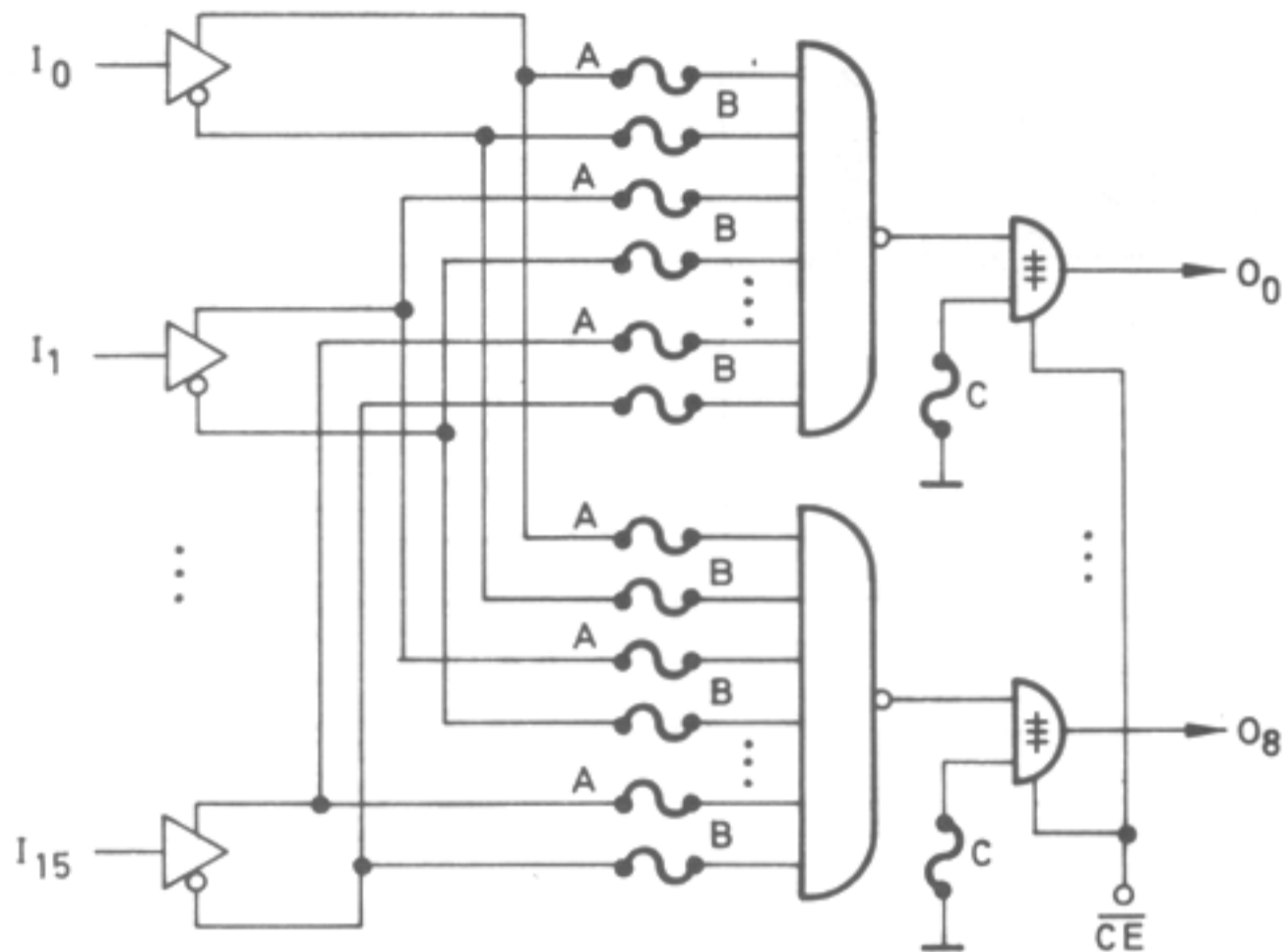


Bild 2.4 Logikdiagramm für ein FPGA

Ober die 16 invertierenden und nichtinvertierenden Eingangsgatter werden insgesamt 32 Variable erzeugt, d.h. für den Anwender stehen jeweils die direkte oder die negierte Variable für die Verknüpfung zur Verfügung. Jede direkte oder negierte Eingangsvariable ist über ein Sicherungselement mit einem NAND-Gatter verbunden. Dadurch hat jedes NAND-Gatter 32 Eingangsvariable.

Die Ausgänge der NAND-Gatter sind über Exklusiv-ODER-Gatter mit den betreffenden Ausgängen verbunden. Hier sind ebenfalls 2 Sicherungselemente vorhanden. Wird das Sicherungselement nicht programmiert, also nicht abgeschmolzen, erscheint die Ausgangsvariable des NAND-Gatters direkt (nicht negiert). Erfolgt aber eine Programmierung, wird die Ausgangsvariable des NAND-Gatters negiert.

Für die freiprogrammierbare Gatter-Anordnung von Bild 2.5 gelten folgende Bedingungen für die Programmierung:

Eingang negiert: Sicherungselement A programmieren

Eingang positiv: Sicherungselement B programmieren

Eingang nichtig: Sicherungselement A und B programmieren

Ausgang H-aktiv: Sicherungselement C programmieren

Ausgang L-aktiv: Sicherungselement C intakt

Die Bezeichnung "Eingang nichtig" bedeutet, daß ein "don't care" vorliegt, also das NAND-Gatter nicht benötigt wird.

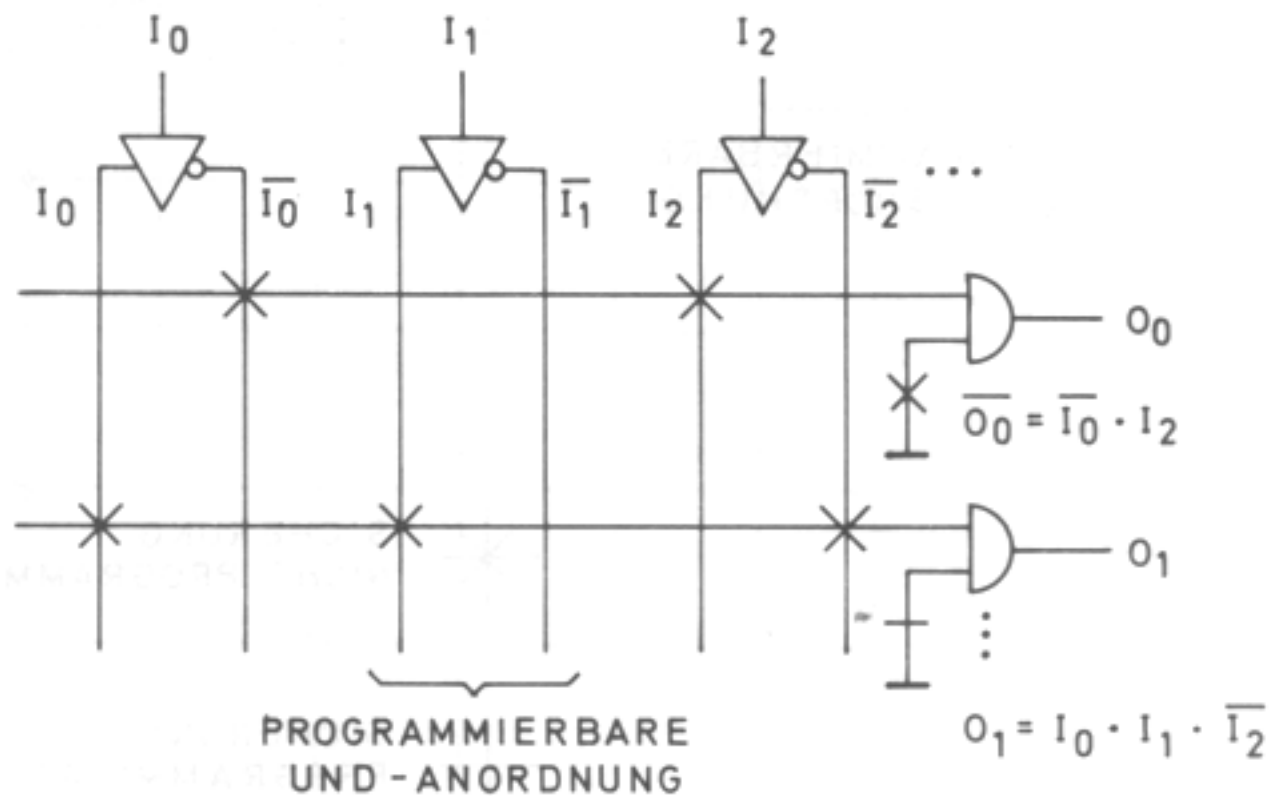


Bild 2.5 Beispiel für die Anwendung eines FPGA

Bild 2.5 zeigt ein Programmierbeispiel für eine programmierbare Gatter-Anordnung. Die drei Eingangsvariablen I_0 , I_1 und I_2 werden zu zwei Eingangsvariablen O_0 und O_1 verknüpft.

2.5 Programmierbare Array-Logik (PAL Programmable Array Logic)

Die "Programmable Array Logic" oder PAL ist eine der neuesten Speicherentwicklungen. Mit den PAL-Bausteinen können direkt TTL-Logiksysteme auf kleinstem Raum realisiert werden. Bild 2.6 zeigt die Anordnung der einzelnen Funktionselemente.

Über ein Eingangsgatter werden jeweils eine direkte und eine negierte Eingangsvariable erzeugt, die dann an den einzelnen Verknüpfungszeilen anliegen. Durch diese Verknüpfungszeilen werden die konjunktiven Terme hergestellt. Am Ende einer Zeile befindet sich ein Buffer und ein ODER-Gatter für die Zusammenfassung der einzelnen Terme. Damit ergibt sich die disjunktive Verknüpfung der Terme.

Im Bild 2.6 ist ein Programmierbeispiel eingezeichnet. Aus den drei Eingangsvariablen wird eine Ausgangsvariable hergestellt.

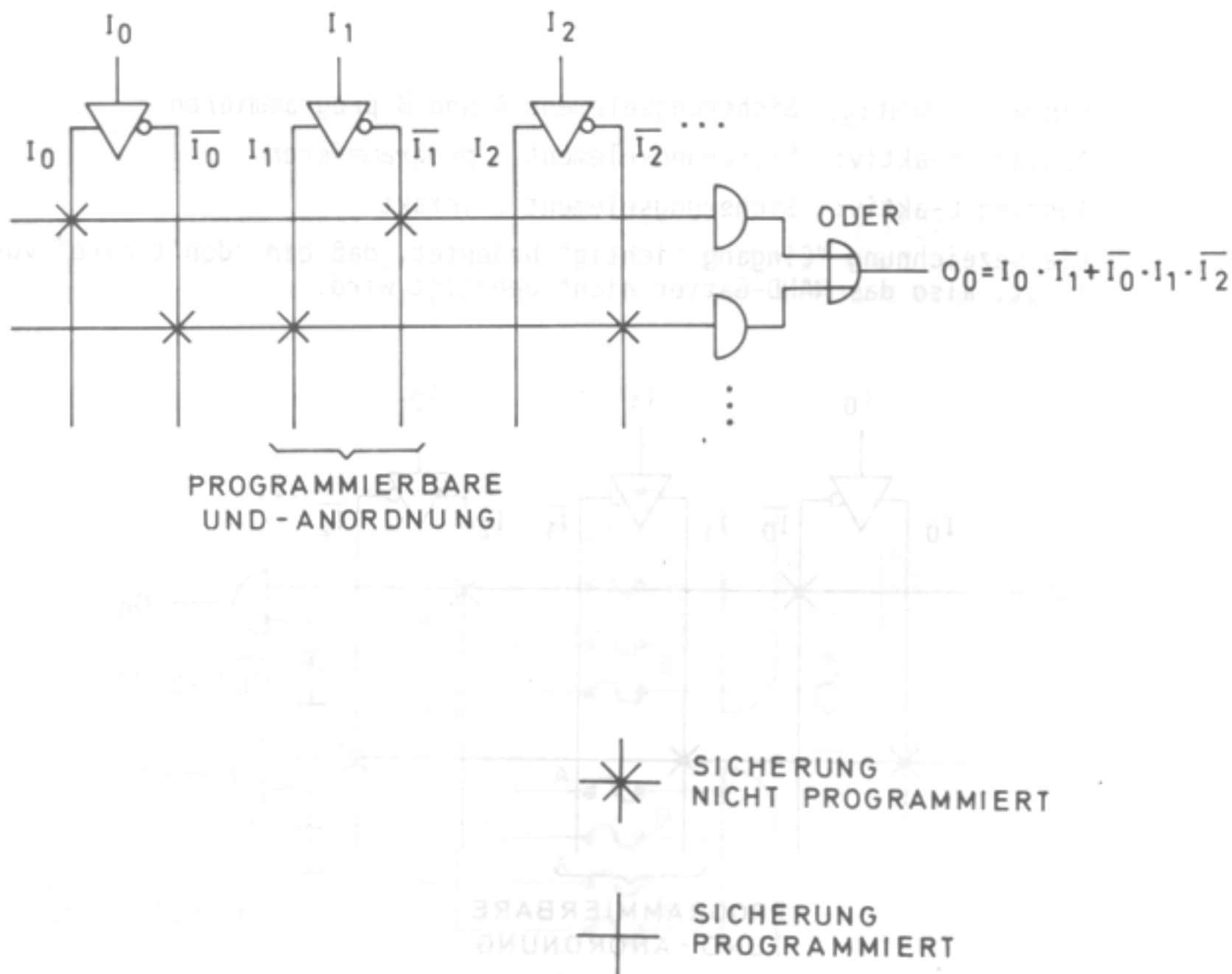


Bild 2.6 Beispiel für eine PAL-Anwendung

Zur Zeit werden ca. 20 verschiedene PAL-Bausteine angeboten.
 Organisation: 10 Eingänge und 8 Ausgänge, 16 Eingänge und 8 Ausgänge,
 mit oder ohne Ausgangsspeicher, offener Kollektor- oder
 Tri-State-Ausgang usw.

Bereits bei einer einfachen Logikanwendung läßt sich der TTL-Baustein-
 aufwand auf ein Minimum reduzieren. Bei der Rückkopplung der Ausgänge
 auf die Eingänge sind Controller, Spezialzähler, komplexe Digitalschal-
 tungen usw. innerhalb eines Bausteines zu realisieren.

2.6 Programmierbare Logik-Anordnung

(FPLA Field Programmable Logic Array)

Im Gegensatz zur PAL-Technik steht das "Field Programmable Logic Array"
 oder FPLA.

Bild 2.7 zeigt den Aufbau eines Speichertyps mit einem Programmierbei-
 spiel. An den Eingängen I liegen drei Eingangsvariable an, die durch
 einen Buffer invertiert oder direkt auf die Termzeilen gegeben werden.
 Durch die Sicherungselemente und Dioden kommt eine Verbindung zwischen
 den Leitungen zustande.

Wird keine Verbindung gewünscht, kann dies durch einen Programmierimpuls erreicht werden. Es ergibt sich eine konjunktive Verknüpfung. Durch Buffer sind das konjunktive und das disjunktive Verknüpfungsfeld voneinander getrennt.

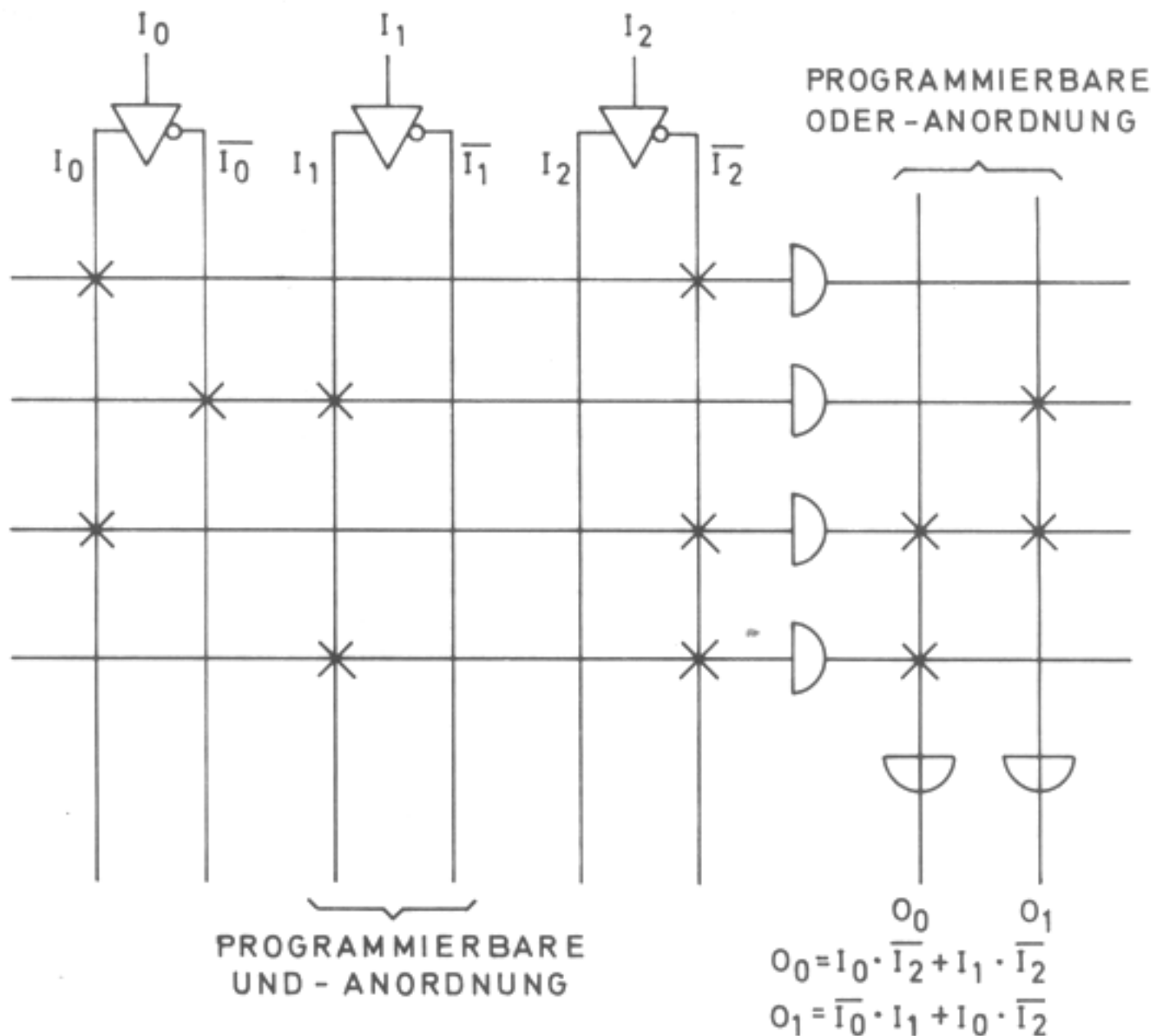


Bild 2.7 Beispiel für eine FPLA-Anwendung

In dem ODER-Feld erfolgt die Verknüpfung der einzelnen Zeilen zu einer Ausgangsfunktion. Auf diese Weise können bis zu 8 Spaltenterme programmiert werden. Durch Ausgangsbuffer liegen die einzelnen Terme an den Ausgängen an.

2.7 Handhabung von MOS-PROMs

Die Handhabung von MOS-PROMs ist mit der gleichen Gültigkeit verbunden, wie dies mit den Standard-Logik-MOS-Schaltkreisen der Fall ist. Bei einer unsachgemäßen Handhabung, z.B. durch Berühren mit elektrostatisch aufgeladenen Körpern, wird der Baustein zerstört !

2.8 Löscheinrichtungen für EPROMs

Die Programmierung von EPROMs bringt keine Schwierigkeiten, wenn die herstellerepezifischen Werte eingehalten werden. Über- oder unterschreitet man diese Werte, kommt es zu Fehlinterpretationen und der

Baustein wird zerstört. Beim Löschen des Speicherinhaltes kann es Probleme geben, wenn ebenfalls nicht die Werte des Herstellers genau eingehalten werden. Häufig wird eine mangelhafte Löschung dem Programmiergerät angelastet, obwohl in Wirklichkeit eine ungenügende Löschung vor der Programmierung die eigentliche Ursache ist. Selbst original verpackte EPROMs sollen vor dem Programmieren gelöscht werden.

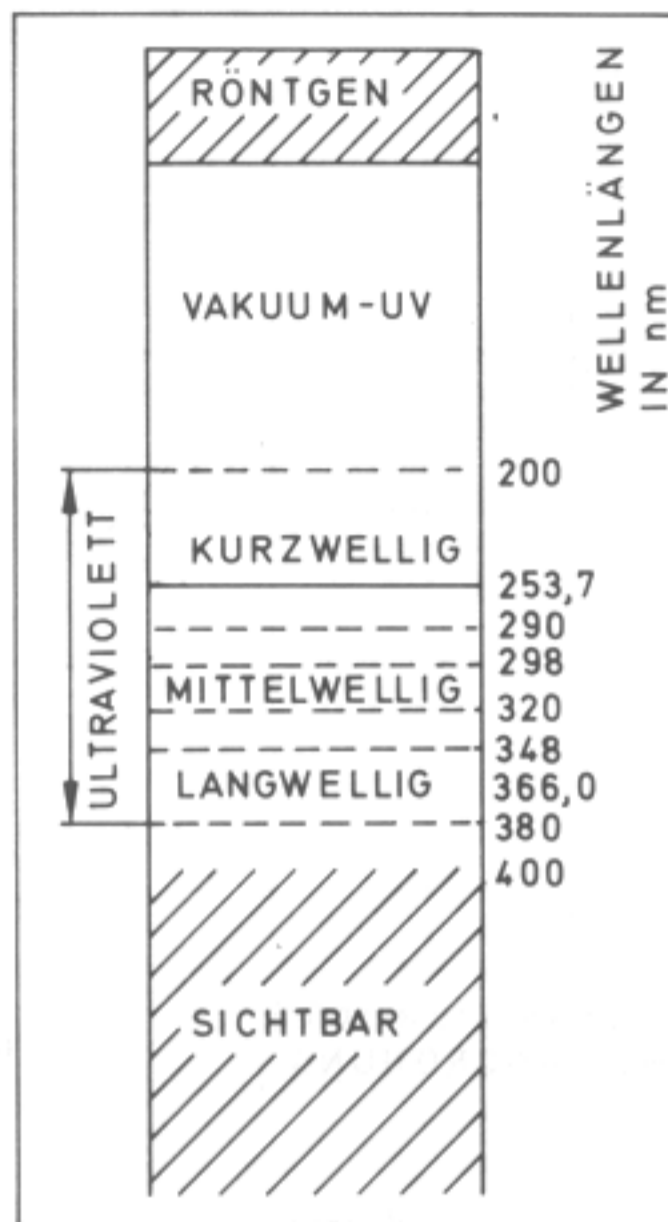


Bild 2.8 Wellenspektrum für EPROM-Löschvorgang

Bild 2.8 zeigt das Wellenspektrum. Die energiereichen UV-Strahlen schließen sich unmittelbar an den Bereich des sichtbaren Lichtes an. Die Löschung der EPROMs erfolgt mit hochintensivem UV-Licht der Wellenlänge 254 nm oder 2540 Å. Auf dem EPROM befindet sich ein Quarzfenster, durch welches das UV-Licht auf das Speicherfeld fällt. Durch die UV-Strahlung kommt es auf dem EPROM-Chip zu einer Ionisation, die zu einem Ladungsausgleich führt, d.h. es tritt eine Informationslöschung auf.

Die richtige Dosierung der UV-Lichtmenge ist ein entscheidender Faktor für eine erfolgreiche Wieder-Programmierung von EPROMs.

2.9 EPROM-Löschtest

Eine einfache Methode zur Ermittlung der erforderlichen Mindestlöschzeit ist im folgenden kurz beschrieben:

- 1) Ein oder mehrere programmierte EPROMs sind in ein EPROM-Löschgerät zu legen und für ca. eine Minute der hochintensiven UV-Strahlung auszusetzen.
- 2) Ein EPROM ist dem ausgeschalteten Löschgerät zu entnehmen und mit dem Leertest (blank check) des Programmer-Systems 19 zu prüfen, ob alle Bit ordnungsgemäß gelöscht sind.
- 3) Ist das EPROM nicht "gelöscht", Löschvorgang mit dem gleichen Zeitintervall wiederholen. Punkt 2 und 3 sind so lange zu wiederholen, bis zum ersten Mal festgestellt wird, daß der gesamte Dateninhalt gelöscht ist. Dies ergibt den Zeitpunkt X für Bild 2.9.

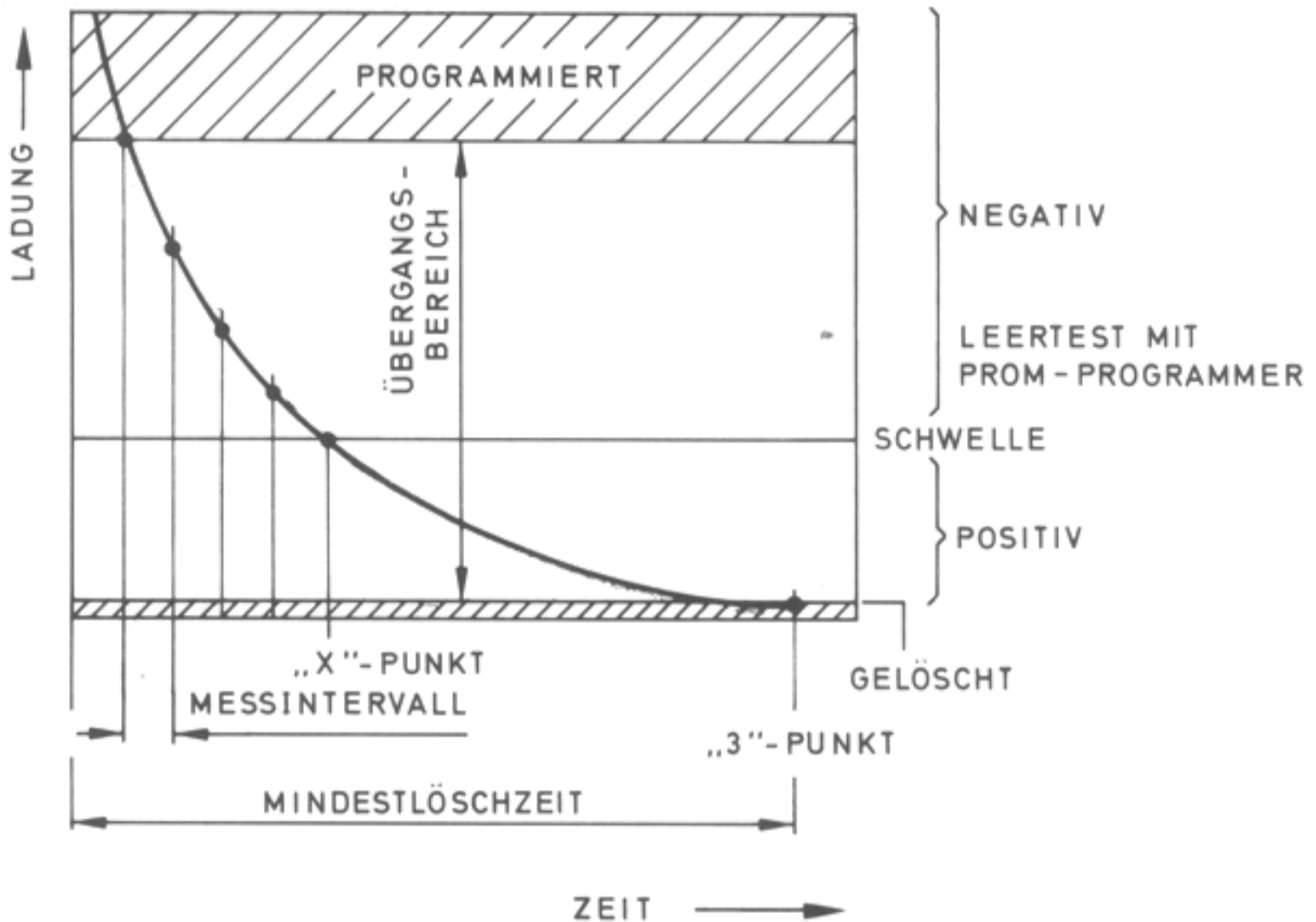


Bild 2.9 Diagramm zur Ermittlung der Mindestlöschzeit

- 4) Um die Mindestlöschzeit zu erhalten, ist die Zeit X mit dem Faktor 3 zu multiplizieren. Das Ergebnis stellt das Minimum der notwendigen Löschzeit bezogen auf die benutzten EPROMs und die Strahlenintensität der verwendeten Löschlampe dar.

Die so ermittelte Mindestlöschzeit kann jedoch nicht als unbedingt gültiger Festwert betrachtet werden, da die Löschzeit von folgenden wichtigen Faktoren abhängig ist:

- 1) Sie ist von dem betreffenden EPROM-Typ abhängig.
- 2) Auch bei gleichen EPROM-Typen ergeben sich Fertigungstoleranzen.

- 3) Die zum Löschen eines EPROM erforderliche Lichtenergie bei wiederholter Löschung und Wiederprogrammierung zunimmt.
- 4) Die Intensität der UV-Lichtquelle mit der Zahl der Betriebsstunden abnimmt.

Um auch langfristig die ausreichende Löschung der EPROMs zu sichern, ist es notwendig, die Ermittlung der Mindestlöschzeit in gewissen Zeitabständen zu wiederholen.

Bild 2.9 zeigt ein Diagramm zur Ermittlung der Mindestlöschzeit für EPROMs, damit eine sichere Programmierung gewährleistet ist.

Die bei EPROMs nach ca. 50 bis 100 Programmzyklen feststellbare merkliche Erhöhung der Mindestlöschzeit läßt sich meistens durch eine Hochtemperaturlagerung bei ca. 200°C über einige Stunden wieder normalisieren.

