

```

*****
*
*
*
*
*          BEDIENUNGSANLEITUNG
*
*
*  UNIVERSAL  PROGRAMMIER-MODUL
*
*
*          U P M
*
*
*  UPM-14M1085-D
*
*****

```

10/85



## Vorwort

Dieses Handbuch beschreibt den Gebrauch und die Handhabung des Universal-Programmier-Moduls UPM. Es beinhaltet sowohl die nötigen Informationen zur Inbetriebnahme auf den Programmern EPP-80 und MPP-80S als auch für die Bedienung generell. Es sind sämtliche erforderlichen Bedienschritte, die zur erfolgreichen Programmierung von EPROMS, PAL's\* und bipolaren Bauteilen nötig sind, beschrieben.

Zur Bedienung der Grundgeräte (MPP-80S bzw. EPP-80) mit den dazugehörigen Eingaben, Befehlen und Meldungen sei auf das entsprechende Bedienerhandbuch verwiesen. Auf die Grundgeräte wird nur soweit eingegangen, wie es für den Betrieb des Programmiermoduls notwendig ist.

\* PAL ist ein eingetragenes Warenzeichen der Monolithic Memories INC.

## REVISION HISTORY

<u>Titel</u>	<u>Nummer</u>	<u>Datum</u>	<u>UPM Version</u>
Universal- Programmier- Modul UPM Bedienungs- anleitung	UPM-13M0785-D	7/1985	X 1.3
Universal- Programmier- Modul UPM Bedienungs- anleitung	UPM-14M1085-D	10/1985	V 1.4

## Weitere Veröffentlichungen

PROM Programmer MPP-80S / EPP-80  
User Reference Manual 5000-01

Selecting Guide for  
Universal Device Programmers

I n h a l t		Seite
<b>1.</b>	<b>ALLGEMEINES.....</b>	<b>6</b>
1.1	Funktionstasten und Displaymeldungen.....	7
<b>2.</b>	<b>INBETRIEBNAHME.....</b>	<b>8</b>
2.1	Automatische Selbsttestroutinen.....	9
2.2	Einsetzen der Speicherbausteine.....	10
<b>3.</b>	<b>AUSWAHL DER BAUTEILETECHNOLOGIE.....</b>	<b>11</b>
3.1	Selektierung von PALs.....	13
3.1.1	Last Fuse Programmierung.....	15
3.1.2	Konvertierungsprogramm .....	16
3.2	Selektierung von EPROMs.....	18
3.2.1	Silicon-Signature.....	19
3.3	Selektierung bipolarer Bauteile.....	20
<b>4.</b>	<b>ZUSÄTZLICHE UPM-FUNKTIONEN.....</b>	<b>23</b>
4.1	PAL-Inhalt in EPROM abspeichern ('MOVE').....	23
4.2	EPROM-Master in Programmer RAM laden ('INSert') .....	25
4.3	CHeCK-Funktionen einschließlich Vektor-Test.....	26
4.4	FC-Funktionen.....	28
4.5	Ausgabe der UPM-internen Bauteileliste.....	30
<b>5.</b>	<b>DATEN EIN-/AUSGABE.....</b>	<b>31</b>
5.1	JEDEC-Datenübertragungsprotokoll.....	31
5.2	Datenformat-Beispiele.....	35
<b>6.</b>	<b>UPM - FEHLERMELDUNGEN.....</b>	<b>43</b>
<b>7.</b>	<b>CHECKSUMMEN-ALGORITHMEN.....</b>	<b>45</b>
7.1	Additive Checksumme.....	45
7.2	KONTRON CRC-Checksumme.....	47
7.3	Programmierbeispiel für Checksummen.....	49

## ANHANG

Discrepancy Report  
Software-Update Vertrag

## 1. ALLGEMEINES

Das mit den Grundgeräten EPP-80 und MPP-80S kompatible Universal-Programmier-Modul UPM bietet dem Anwender hohen Programmierkomfort und gewährleistet dabei durch die vielfältigsten Überprüfungs-routinen ein Höchstmaß an Sicherheit. Für viele Anwender erübrigt sich durch den Einsatz des UPM-Moduls die Anschaffung zusätzlicher Module.

Mit dem UPM lassen je nach Version (UPM/A, UPM/B, UPM/C) folgende **Technologien** programmieren:

	UPM/A	UPM/B	UPM/C
EPROM (MOS/CMOS)	x	x	x
EEPROM	x	x	x
BIPOLAR (FL/AIM)	x	x	-
PAL	-	x	x
PAL-ECL	-	x	x
PAL-CMOS	-	-	x
PAL 40-pin	-	-	x

Bauteile der folgenden Hersteller sind implementiert:

E(E) PROM	BIPOLAR	PAL	
AMD	AMD	Altera	*
EA	Fairchild	AMD	
Exel	Fujitsu	Cypress	
Fairchild	Harris	Harris	
Fujitsu	Hitachi	Lattice	*
GI	Intersil	MMI	
Hitachi	Mitsubishi	NS	
Hughes	MMI	TI	
Intel	Motorola		
Intersil	NEC		
Mitsubishi	NS		
Mostek	Raytheon		
Motorola	Signetics		
NS	TI		
NEC			
OKI			
RCA			
Ricoh			
Rockwell			
Seeq			
SGS Ates			
Signetics			
Synertec			
Thomson-EFCIS			
TI			
Toshiba			
Xicor			

\* In Vorbereitung

**Programmierbare Bauteilegrößen mit Angabe des min./max. Array-Umfanges:**

E(E)PROM	BIPOLAR	PAL
1K x 8	32 x 8	Alle Typen
bis	bis	inklusive product term sharing und Polarity-Bauteile
64K x 8	8K x 8	

Eine detaillierte Aufstellung aller mit dem UPM-Modul programmierbaren Bauteiltypen finden Sie im KONTRON "Selecting Guide for Universal Device Programmers", den Sie auf Anfrage gerne von uns erhalten. Sie können die Liste jedoch auch aus dem UPM-Modul direkt erhalten. Über die Output-Funktion 99 (siehe auch Kapitel "Zusätzliche UPM-Funktionen") kann die Liste der programmierbaren Bauteile über eine angeschlossene Schnittstelle ausgegeben werden.

## 1.1 Funktionstasten und Displaymeldungen

Sämtliche in diesem Handbuch beschriebenen Displaymeldungen und Tasteneingaben entsprechen dem EPP-80 Grundgerät. Beim MPP-80S können bei Versionen mit 7-Segment-Anzeige aufgrund des etwas eingeschränkten Displays nicht alle aufgeführten Zeichen in derselben Form dargestellt werden. Neuere Modelle MPP-80S verfügen jedoch über eine alphanumerische Anzeige und sind mit dem EPP-80 Grundgerät kompatibel.

Manche Eingaben beim MPP-80S und EPP-80 weichen voneinander ab und sind wie im Manual beschrieben zu behandeln, z.B.:

EPP-80:     # A - Taste   separate Taste auf dem 12-er Tastenblock  
MPP-80S:     A - Taste   vom Hex-Key-board

Beim MPP-80S Programmiergerät zeigt die 6. Stelle des Displays an, in welcher Betriebsart sich das Gerät befindet:

READY^FOR XXXX	Compressed Mode aktiviert
READY-FOR XXXX	Split/Shuffle Mode aktiviert
READY=FOR XXXX	Sowohl Compressed Mode als auch Split/Shuffle aktiviert

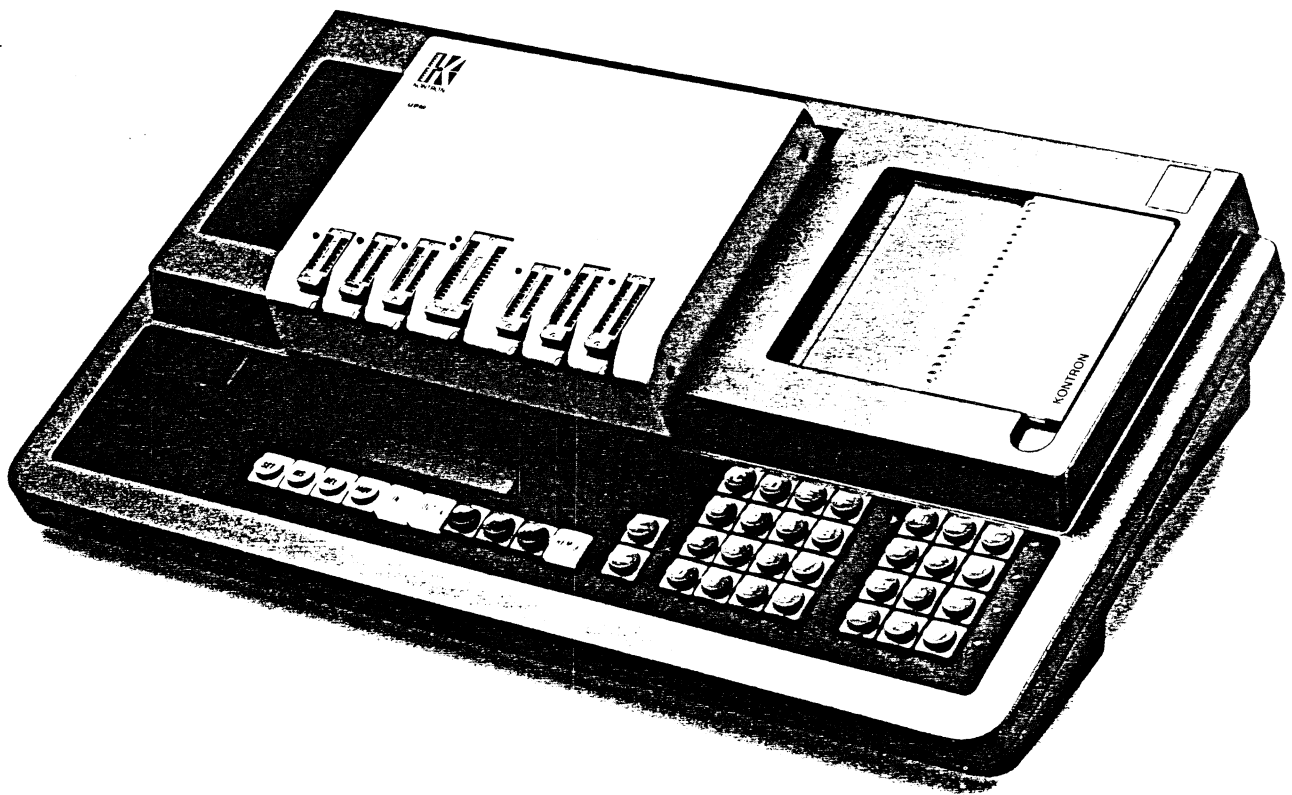
Beim EPP-80 wird die Betriebsart durch die LED neben der jeweiligen Funktionstaste angezeigt.

## 2. INBETRIEBNAHME

Vor dem Einschalten Ihres Programmiergerätes (MPP-80S oder EPP-80) muß das Programmiermodul aufgesetzt werden. Achten Sie darauf, daß in den an der Unterseite des UPM angebrachten Steckerleisten keine Kontakte verbogen sind.

Das UPM kann nur in eine Richtung aufgesetzt werden.

Einschalten des Grundgerätes ohne aufgesetztem Modul führt zur Anzeige undefinierbarer Zeichen, das Gerät reagiert auf keine Eingabe. Nachträgliches Einsetzen des Moduls ändert diesen Zustand nicht. Das Gerät muß erst aus- und, nachdem das Modul eingesetzt wurde, wieder eingeschaltet werden (Reset).



EPP-80 mit UPM

## 2.1 Automatische Selbsttestroutinen

Bei korrekt aufgestecktem Programmiermodul erscheinen während der Initialisierungsroutinen unleserliche Zeichen auf dem Display, wie es auch der Fall ist, wenn das UPM nicht aufgesteckt ist.

Anschließend meldet sich der Programmier mit

**HARD TEST      0X**

Ein umfangreicher Hardware-Test prüft alle wichtigen Funktionen und Parameter und stellt die Betriebsbereitschaft des Moduls sicher. Im Falle eines Fehlers werden entsprechende Codes ausgegeben:

**E HARD TEST    xxx**

Falls sich während des Hardware-Tests in einem der Sockel ein Bauteil befindet, wird die Meldung

**REMOVE      DEVICE**

ausgegeben. Nach Entfernen des Bauteils ist der Programmier aus-/einzuschalten, und der Test startet erneut. Nach erfolgreichem Test erscheint in der Anzeige

**RAM TEST ---- 01**

falls ein Speicher größer 32 Kbyte installiert ist (Extension). Die getesteten Segmente werden jeweils mit 01, 02 etc. angezeigt. Nach fehlerfreier Prüfung wird ein

**-- MEMORYTEST --**

durchgeführt. Dabei wird die gesamte Firmware auf Fehlerfreiheit getestet.

Falls die Meldung

**E    CrC**

auftritt, deutet das auf einen Speicherfehler im Gerät hin.

Das Programmiergerät kann dann nicht benutzt werden und ist umgehend zur Überprüfung einzuschicken.

Wurden alle Testroutinen erfolgreich durchlaufen, zeigt der Programmier mit der Meldung

**SELECT EPROM**

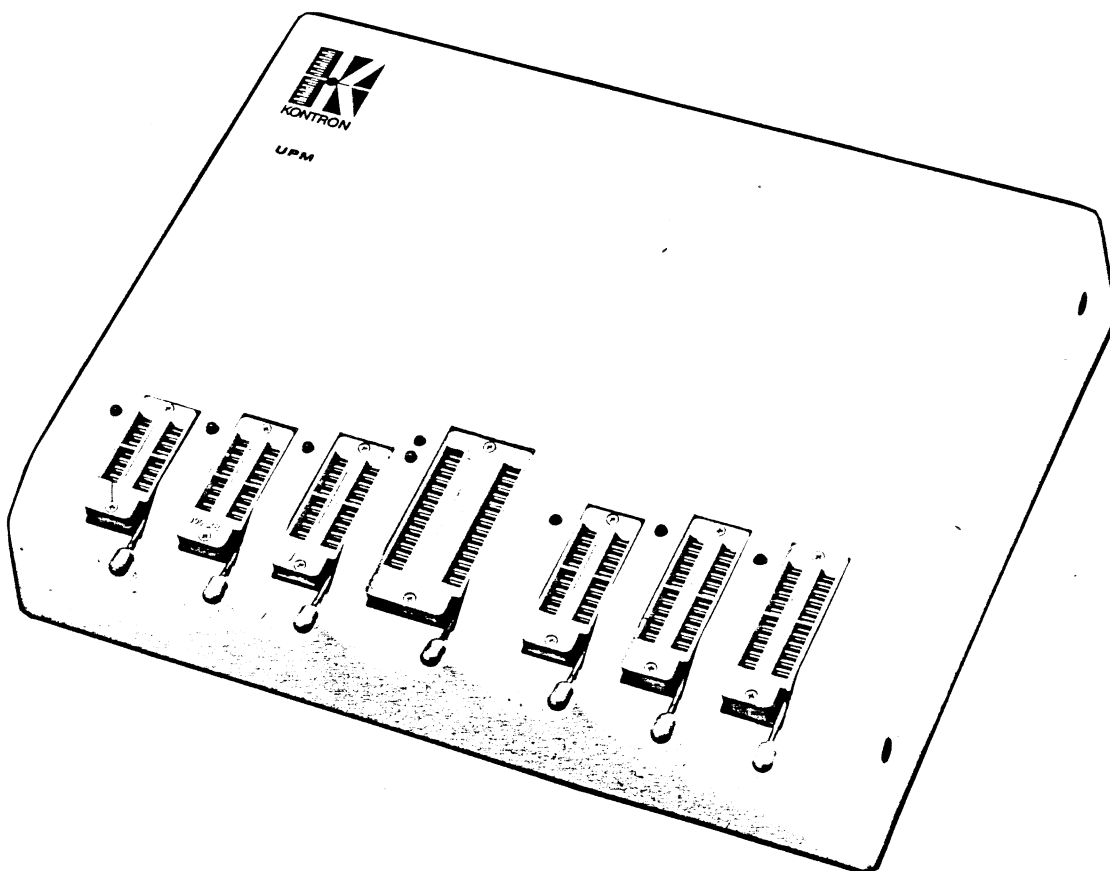
und einem Piepston an, daß nun die Bauteile-Technologie angewählt werden kann.

## 2.2 Einsetzen der Speicherbausteine

Sobald ein Speichertyp ausgewählt wurde, wird eine der Leuchtdioden auf der Oberseite des UPM aktiviert. Diese Leuchtdiode zeigt Ihnen, welcher Sockel zu benutzen und wie der Speicherbaustein einzusetzen ist. Die Leuchtdiode zeigt auf Pin 1 des Speicherbausteins.

Falls ein Speicherbaustein falsch eingesteckt wurde, so wird bei allen Operationen, die auf das Bauteil zugreifen (LOAD, PROG, CHK) eine der folgenden Fehlermeldungen ausgegeben:

	ERROR REVERSE	Bauteil verkehrt eingesteckt
oder	ERROR NO DEV	Bauteil nicht oder nicht korrekt eingesteckt
oder	ERROR SOCKET	Bauteil im falschen Sockel



UPM - Modul

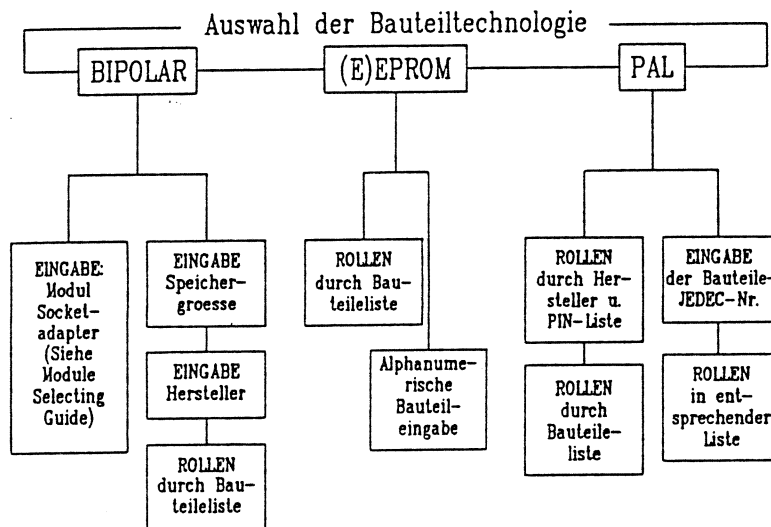
Die IC-Fassungen sind entriegelt, wenn der Hebel nach oben weist und werden durch Umlegen des Hebels nach unten verriegelt.

## 3. AUSWAHL DER BAUTEILETECHNOLOGIE

Das UPM erlaubt je nach Version, zwischen verschiedenen Speichertechnologien zu wählen:

- EPROM (MOS und CMOS E(E)PROMS)
- PROM (Bipolar)
- PAL (Programmierbare Logik)

Für jede Technologie gibt es mindestens zwei verschiedene Auswahlmöglichkeiten, die von den jeweils bekannten Parametern abhängen. Das UPM kann somit auch ohne Zuhilfenahme von Auswahllisten oder ähnlichem auf den gewünschten Baustein eingestellt werden.



Während sich das Gerät im Auswahlmodus befindet (der Schriftzug **SELECT** blinkt), können Sie mit der " - " -Taste durch das Auswahlmenü rollen.

Wenn Sie die gewünschte Speichertechnologie angewählt haben, bestätigen Sie mit der ENTER - Taste.

! Eingabe	Display	!
! -	SELECT EPROM	!
! -	SELECT PROM	!
! -	SELECT PAL	!
! -	SELECT EPROM	!
!	.	!
!	.	!
!	.	!

## Bedienungsanleitung UPM

Unter jeder Speichertechnologie können nun die verschiedensten Bauteiltypen angewählt werden.

Der Auswahlmodus ist auch hier entsprechend:

Auswahl mit der " - " -Taste

Bestätigen mit der ENTER - Taste

Falls Sie sich vertippt haben, kommen Sie mit der 'A' -Taste (MPP-80S) bzw. der #A - Taste (EPP-80) wieder in den entsprechenden Auswahlmodus. Sind Sie bereits in einem Untermenü (z.B. in der Herstellerliste der bipolaren Speicher), muß, um in den Auswahlmodus zurückzukehren, die #A - Taste zweimal gedrückt werden.

## 3.1 Selektierung von PAL's (nur UPM/B und UPM/C)

Mit dem Programmiermodul können PALs im 20- oder 24-poligen Gehäuse programmiert werden.

Der gewünschte PAL-Typ kann auf zwei verschiedene Arten eingestellt werden:

### Erste Möglichkeit:

Rollen durch Hersteller- und Pin-Liste.

Taste	Anzeige	Bedeutung
	SELECT EPROM	Standard-Einstellung
-	SELECT PROM	
-	SELECT PAL	
ENTER	SELECT MMI 20/A	PAL-Technologie wurde ausgewählt und mit der ENTER-Taste bestätigt. Mit der '-' Taste kann jetzt durch eine Liste gerollt werden, in der Hersteller und Pinout gespeichert sind.
-	SELECT MMI 24/A	PAL - Hersteller MMI und 24-Pin Bauteile sind ausgewählt.
ENTER	DIAL TYPE 12L10	Rollen Sie nun mit der '-' Taste durch die Liste der verfügbaren 24-Pin PAL-Bauteile bis der gewünschte Typ in der Anzeige erscheint.
-	DIAL TYPE 14L8	
ENTER	-- MEMORYTEST --	Der angezeigte Speichertyp wird mit der ENTER-Taste bestätigt. Der Programmierer führt einen Speichertest durch und transferiert das Leer-Muster des PALs in das RAM.
	READY MMI 14L8	Das Programmiermodul ist jetzt zur Programmierung des Speicherbauteils 14L8 eingestellt (die Betriebsart "Compressed" ist automatisch eingestellt). Die LED am entsprechenden Sockel leuchtet.

## Zweite Möglichkeit:

Nach Anwahl der PAL-Technologie können Sie den gewünschten Typ durch die Eingabe des PAL-Jedec-Codes gleich direkt einstellen, wobei es keine Rolle spielt, welcher Hersteller gerade angezeigt wird. Dieser Code, der im KONTRON "Selecting Guide for Universal Device Programmers" oder in der UPM-internen Bauteileliste zu finden ist, wird über die Hex-Tastatur eingegeben.

Falls mehrere PAL's unter derselben JEDEC-Nummer existieren (der Jedec-Code 9724, zum Beispiel, steht für die Speicherbauteile AMD 16R4, 16R6 und 16R8), können Sie in einem Untermenü mit der '-' Taste zu dem gewünschten Typ rollen.

Taste	Anzeige	Bedeutung
	SELECT PAL	PAL-Technologie ist angewählt
ENTER	SELECT MMI 20/A	Das Modul ist momentan auf 20-Pin Bausteine eingestellt. Welcher Hersteller gerade angezeigt wird, ist abhängig von den vorherigen Eingaben und hat keinen Einfluß auf die Eingabe der Jedec-Nummer.
9725	DIAL TYPE 16H8	Geben Sie nun den Jedec-Code für den gewünschten Speichertyp (hier: AMD 16H8) über die Hex-Tastatur ein.
ENTER	-- MEMORYTEST --	Speichertest wird durchgeführt.
	READY AMD 16H8	Das Modul ist für die Programmierung des AMD PAL 16H8 eingestellt. Die LED am entsprechenden Sockel leuchtet.

### 3.1.1 Last Fuse Programmierung

PAL-Bausteine bestimmter Hersteller verfügen über eine Kopierschutzeinrichtung, sogenannte "Security Fuse(s)" oder "Last Fuse(s)". Diese letzten Sicherungen können nach dem eigentlichen Programmiervorgang programmiert werden, wodurch verhindert wird, daß das Programmierarray dieses Bausteines wieder gelesen und kopiert werden kann. Die logische Funktion des PALs bleibt dabei erhalten.

Für eine erfolgreiche Last Fuse-Programmierung müssen folgende Bedingungen erfüllt sein:

- a) das Programmierarray darf nicht leer sein,
- b) der gewählte Baustein muß Security Fuse(s) haben,
- c) der eigentliche Programmiervorgang muß fehlerfrei abgeschlossen sein,
- d) das G-Feld im JEDEC File muß eine '1' haben, oder die Security Fuse - Funktion muß mit FC 32 aktiviert sein.

Taste	Anzeige	Bedeutung
	READY AMD 16H8	Das PAL AMD 16H8 wurde zur Programmierung gewählt.
PRO	PR P >	PAL einsetzen
ENTER	--- BUSY --- PR P > XXXX ---VERIFY---	Programmiervorgang läuft, die jeweils programmierten Adressen werden angezeigt.
	PR LA FUSE	Programmer fragt, ob die Kopierschutzsicherung(en) programmiert werden soll(en).
PRO	--BLANK CHECK-- LA FUSE PASS	Die letzte Sicherung des Arrays wurde erfolgreich programmiert.
	oder	
	LA FUSE FAIL	Programmierung der letzten Sicherung ist nicht möglich.

Nach erfolgreicher Security Fuse Programmierung wird beim Versuch, das PAL zu laden, je nach PAL-Type entweder ein leeres Array in das RAM gelesen oder die Meldung

ERROR LA FUSE

ausgegeben.

**3.1.2 Konvertierungsprogramm**

Das UPM hat für (untereinander) kompatible PALs ein automatisches Konvertierungsprogramm eingebaut, mit dem der Inhalt eines PALs auf ein kompatibles PAL eines anderen Herstellers und umgekehrt umgewandelt werden kann. Angenommen, es wird das Muster eines MMI PALs geladen und anschließend der äquivalente Typ eines anderen Herstellers (z.B. AMD) gewählt um eine Kopie herzustellen, dann wird das im RAM befindliche Programmierarray auf das Programmierformat von AMD umgewandelt. Die Konvertierung läuft wie folgt ab:

Taste	Anzeige	Bedeutung
	READY MMI 16L8	Anzeige des aktivierten PALs.
	.	
	.	
-	SELECT AMD 20	Der neue PAL-Typ (20 -Pin) wird ausgewählt
	.	
	.	
ENTER	DIAL TYPE 16L8	Der gewünschte Typ wurde gefunden und mit ENTER bestätigt.
ENTER	- FUSE CONVERT -	Das Fuse Pattern des Bauteils 16L8 von MMI wird auf ein funktionsgleiches PAL von AMD konvertiert. Die Testvektoren bleiben unverändert erhalten.
	READY AMD 16L8	

**Anm.:** Soll von einem PAL eines Herstellers auf ein funktionsgleiches PAL eines anderen Herstellers umgeschaltet werden ohne daß eine Konvertierung gewünscht wird, muß zwischen den beiden Typen ein anderer Typ selektiert werden. Dadurch wird das vorhandene Programmierarray und die Testvektoren gelöscht. Sind bei funktionsgleichen PALs auch die Programmierarrays identisch, so erfolgt kein Fuse Convert.

## UPM Konvertierungstabelle für PAL Bauteile:

Konvertierung von Hersteller zu Hersteller	!	Typ	!	Änderungen im RAM
	!		!	
MMI/NS/TI <--> AMD/HAR/CYP	!	gleich	!	Konvertierung aktiv
MMI/NS/TI <--> AMD/HAR/CYP	!	verschieden	!	RAM und Test- vektoren gelöscht
	!		!	
MMI/NS/TI <--> MMI/NS/TI	!	gleich	!	RAM unverändert
MMI/NS/TI <--> MMI/NS/TI	!	verschieden	!	RAM und Test- vektoren gelöscht
	!		!	
AMD/HAR/CYP <--> AMD/HAR/CYP	!	gleich	!	RAM unverändert
AMD/HAR/CYP <--> AMD/HAR/CYP	!	verschieden	!	RAM und Test- vektoren gelöscht
	!		!	

## 3.2 Selektierung von EPROMs

Kehren Sie durch zweimaliges Drücken der '#A'-Taste in das "SELECT EPROM / PROM / PAL" Auswahlmenü zurück und wählen Sie, wie im Kapitel 2.1 beschrieben, EPROM-Technologie an. Wie bei der PAL-Programmierung gibt es auch hier zwei Möglichkeiten, das Modul auf den gewünschten Bauteiletyp einzustellen.

### Erste Möglichkeit:

Rollen durch Bauteile-Liste.

Taste	Anzeige	Bedeutung
	SELECT EPROM	EPROM - Technologie ist angewählt.
ENTER	DIAL TYPE 2716	EPROM - Technologie wird bestätigt und Grundeinstellung angezeigt.
-	DIAL TYPE xxxx	Rollen Sie mit der '-' Taste durch die Liste der verfügbaren Bauteile bis der gewünschte Typ angezeigt wird und bestätigen Sie dann mit 'ENTER'.
.		
.		
.		
.		
-	DIAL TYPE 27128F	
ENTER	READY FOR 27128F	Das Modul ist für die Programmierung des gewählten Bausteines eingestellt. Die LED am entsprechenden Sockel leuchtet.

### Zweite Möglichkeit:

Alphanumerische Eingabe des gewünschten EPROM-Typs.

Taste	Anzeige	Bedeutung
	SELECT EPROM	EPROM-Technologie ausgewählt
ENTER	DIAL TYPE 2716	Standardeinstellung
2732A	DIAL TYPE 2732A	Geben Sie den gewünschten Bauteiltyp direkt über die Hex-Tastatur ein.
ENTER	READY FOR 2732A	Das Modul ist auf den angezeigten Typ eingestellt.

### 3.2.1 Silicon Signature

Wenn Sie Speicherbauteile mit 'Silicon Signature' verwenden, können Sie die Einstellung auf den betreffenden Typ gänzlich dem UPM - Modul überlassen. Befindet sich im EPROM-Modus ein EPROM mit 'Silicon Signature' im Sockel, erkennt das Modul bei Betätigen der #A - Taste (EPP-80) bzw. A - Taste (MPP-80S) automatisch den eingesetzten Baustein.

Taste	Anzeige	Bedeutung
	READY FOR XXXX	Grundstellung im EPROM-Modus. Es spielt keine Rolle, welches Bauteil gerade selektiert ist.
# A	READY FOR 27256	Das im Sockel befindliche PROM 27256 wurde erkannt und das Modul entsprechend konfiguriert. Die LED am entsprechenden Sockel leuchtet.

Hat das eingesetzte EPROM keine Silicon Signature, erscheint nach Betätigen der A-Taste in der Anzeige

DIAL TYPE 2716

und der gewünschte Typ muß mit einer der beiden anderen Möglichkeiten eingegeben werden.

### 3.3 Selektierung bipolarer Bauteile

Kehren Sie durch Betätigen der #A-Taste in das 'SELECT EPROM / PROM / PAL' Auswahlmenü zurück und wählen Sie, wie im Kapitel 3.1 beschrieben, die PROM - Technologie an. Die Auswahl kann wieder auf zwei Arten vorgenommen werden:

#### Erste Möglichkeit:

Auswahl des Bauteiles über die Eingabe der Personality-Modul (MOD)- Nummer sowie der Sockeladapter (SA) - Nummer aus der Auswahlliste. Die entsprechenden Daten entnehmen Sie bitte dem KONTRON "Selecting Guide for Universal Device Programmers" oder der UPM-internen Bauteileliste.

Z.B.: für Texas Instruments      TI 74S287  
       laut Tabelle:                    MOD 5B      SA 4-2

Taste	Anzeige	Bedeutung
	SELECT PROM	PROM - Technologie ist ausgewählt.
ENTER	SELECT MOD SA	Die Anzeige fordert die direkte Eingabe der Modul- und Sockeladapter-Nummer aus dem 'Module Selecting Guide' an.
5B	MOD 5B	Modul 5B wurde ausgewählt
ENTER	MOD 5B SA	Das Modul erwartet nun die Eingabe der Sockeladapter-Nummer.
4-2	MOD 5B SA 4-2	Sockeladapter 4-2 ist ausgewählt.
ENTER	TI TBP14S10	Sie befinden sich nun in einer Auswahlliste, in der alle Bauteile gespeichert sind, die mit dem Modul MOD 5B und dem Sockeladapter SA 4-2 programmiert werden können. Rollen Sie mit der '-' Taste zum gewünschten Typ.
-	TI TBP14SA10	
-	TI 74S287	Das Speicherbauteil 74S287 von Texas Instruments ist selektiert.
ENTER	TI 74S287	Das Modul ist auf den angezeigten Bauteiletyp eingestellt. Die LED am entsprechenden Sockel leuchtet.

## Zweite Möglichkeit:

Die Auswahl des Bipolar-PROMs erfolgt über die Anwahl des Herstellers und der Speicherkonfiguration (Array Size). Die genauen Daten hierfür liefert Ihnen entweder das jeweilige Datenblatt, der KONTRON "Selecting Guide for Universal Device Programmers" oder die UPM-interne Bauteileliste.

Taste	Anzeige	Bedeutung
	SELECT MOD SA	Standardvorgabe
-	SELECT AMD	Rollen Sie mit der '-' Taste durch die Liste der verfügbaren Hersteller und bestätigen Sie den gewünschten Hersteller mit der ENTER-Taste.
-	SELECT FAIRCHILD	
ENTER	SELECT 256-8	Fairchild ist ausgewählt. Nun können Sie entweder die angezeigte Bauteile-Organisation bestätigen oder mit der '-' Taste zu einer anderen Organisation rollen.
-	SELECT 512-4	Speicherorganisation 512 x 4 wurde angewählt.
ENTER	FAI 93436	Nach Bestätigung der Speicherorganisation können Sie durch die Liste aller Bauteile dieses Herstellers mit der gewünschten Organisation rollen.
-	FAI 93446	Das Speicherbauteil 93446 des Herstellers Fairchild ist angewählt.
ENTER	FAI 93446	Das UPM ist jetzt auf den angezeigten Baustein eingestellt. Die LED am entsprechenden Sockel leuchtet.

## Bedienungsanleitung UPM

Alternativ dazu kann die gewünschte Speicherorganisation aber auch direkt eingegeben werden:

Taste	Anzeige	Bedeutung
	.	
	.	
	.	
	SELECT 256-8	Geben Sie die gewünschte Speicherkonfiguration direkt über die Hex-Tastatur ein.
512-8	SELECT 512-8	Speicherkonfiguration 512 x 8 ist ausgewählt.
ENTER	FAI 93438	Rollen Sie jetzt wieder durch die Liste aller verfügbaren Bauteile von Fairchild mit einer Speicherorganisation von 512 x 8.

#### 4. ZUSÄTZLICHE UPM - FUNKTIONEN

Das UPM bietet dem Anwender eine Reihe zusätzlicher Funktionen, die nachstehend behandelt werden:

- 4.1 PAL-Inhalt in EPROM abspeichern (MOVE-Kommando)
- 4.2 EPROM-Master in Programmer-RAM laden (INSERT-Kommando)
- 4.3 CHECK-Funktionen einschließlich Vektor-Test
- 4.4 FC - Funktionen
- 4.5 Ausgabe der UPM-internen Bauteileliste

##### 4.1 PAL-Inhalt in EPROM abspeichern (MOVE-Kommando)

Das UPM bietet die Möglichkeit, den bereits im RAM befindlichen PAL-Inhalt zusammen mit Testvektoren und einem frei wählbaren, maximal 6-stelligen Identifikationscode in einem EPROM abzulegen.

Dieses EPROM dient dann als Programmiermaster. Jedesmal, wenn der Inhalt dieses Masterproms in das RAM geladen wird, werden automatisch alle Systemparameter geladen und eingestellt.

Taste	Anzeige	Bedeutung
	READY MMI 10H8	Anzeige des PALs, dessen Inhalt in einem EPROM abgespeichert werden soll.
MOV	SELECT 27^	Geben Sie den für den Backup verwendeten Bauteiltyp durch Vervollständigung der angezeigten Typen-Nummer ein. Folgende Bauteile können verwendet werden: 2716 / 2732 / 2732A / 2764 / 2764A (nur Intel oder programmierkompatible Bauteile.

Falls kein EPROM eingesteckt ist, wird am Display beim Betätigen der 'MOV'-Taste die Meldung LD P> (Lade PROM) erscheinen. Beide Leuchtdioden am 24/28-Pin Sockel blinken.

Taste	Anzeige	Bedeutung
16	SELECT 2716	EPROM 2716 wird für den Backup verwendet
ENTER	SAVE ID :	Eine bis zu 6-stellige Identifikation kann über die Hex-Tastatur eingegeben werden (0-9, A-F).
ABC	SAVE ID : ABC	'ABC' wurde als Identifizierung eingegeben.
ENTER	--- BUSY --- -- MEMORYTEST -- SAVE xxxx	Der PAL-Inhalt wird in das gewählte EPROM abgespeichert, wobei die jeweils bearbeitete Adresse angezeigt wird.
	READY MMI 10H8	PAL-Inhalt ist komplett abgespeichert.

## 4.2 EPROM-Master in Programmer RAM laden (INSert-Kommando)

Die mit dem MOVE-Kommando im Master-EPROM gespeicherten PAL-Daten können zur Programmierung weiterer PAL-Bausteine herangezogen werden. Dazu muß zunächst die PAL-Technologie angewählt sein (Kap. 2.1). Dabei spielt es keine Rolle, welcher PAL-Typ gerade ausgewählt ist, da sämtliche Identifikationsdaten im EPROM mitgespeichert sind. Der Ladevorgang arbeitet wie folgt:

Taste	Anzeige	Bedeutung
	READY AMD 16L8	
Befindet sich kein EPROM im Sockel, wird bei Betätigen der INS-Taste die Meldung LD P> (load PROM) angezeigt. Beide LEDs am 24/28-Pin Sockel leuchten.		
INS	--LOAD DEVICE-- LOAD ID: ABC	Zur Kontrolle daß der richtige EPROM-Master mit den gewünschten Daten abgespeichert wird, wird der Identifikationscode, hier ABC, mit ausgegeben. Die momentan gespeicherten Daten (Programmierarray und Testvektoren) sind noch vorhanden.
(-)		Mit der Minus-Taste kann die Funktion abgebrochen werden, mit ENTER wird sie fortgesetzt.
ENTER	--- BUSY ---	Ladevorgang wird ausgeführt. Die Daten werden aus dem EPROM in den Datenspeicher kopiert und ein neues Programmierarray aufgebaut.
	-- MEMORYTEST --	
	Ld P < xxxx	JEDEC Fuse Checksumme wird angezeigt.
ENTER	READY MMI 10H8	Das UPM ist jetzt für die Programmierung des gewählten Bausteins (MMI 10H8) vorbereitet.

## 4.3 Check - Funktionen einschließlich Vektor-Test

Zusätzlich zu den mit dem UPM durchführbaren Check-Arten 0 - 4, die im EPP-80/MPP-80S Bedienerhandbuch ausführlich beschrieben sind, können mit dem Universal-Programmiermodul UPM folgende PAL-spezifische Checks durchgeführt werden:

### CHECK 7/8

Check 7 berechnet die JEDEC Fuse-Checksumme des momentan im Sockel befindlichen PALs. Check 8 berechnet die JEDEC Fuse Checksumme aus dem aktuellen RAM-Inhalt. Beide Checksummen werden als 4-stellige Hex-Zahl im Programmer-Display angezeigt. Die Summe ist identisch mit der im JEDEC File im C-Feld dargestellten Summe. Ein leeres PAL sowie ein PAL mit programmierter Security Fuse hat immer Checksumme 0.

Taste	Anzeige	Bedeutung
	READY MMI 10H8	Das PAL MMI 10H8 ist selektiert.
CHK	CH 00	Programmer erwartet Eingabe der gewünschten Check-Art.
7	CH 07	Check-Art 7 ist gewählt.
ENTER	CH P >	PAL einsetzen.
ENTER	-- BUSY -- -- MEMORYTEST --	Check läuft.
	CHESU XXXX	Die JEDEC Fuse Checksumme des momentan im Sockel befindlichen PALs wird als 4-stellige Hex-Zahl angezeigt.
ENTER	READY MMI 10H8	

Die Eingabeprozedur für Check 8 ist identisch mit Check 7. Es entfällt jedoch der P> - Schritt, da die Checksumme aus dem RAM berechnet wird.

**CHECK 9 - Vektor-Test:**

Wurden mit dem JEDEC - Datenübertragungsprotokoll oder bei der INSERT - Funktion Testvektoren übertragen, so können diese an das programmierte, im Sockel befindliche PAL angelegt werden. Dadurch wird die Funktionssicherheit des Bauteiles im Sockel sichergestellt. Die beim Check 9 übertragenen Vektoren werden sequentiell, entsprechend der im JEDEC V-Feld enthaltenen Nummer an das PAL angelegt. Nicht übertragene Vektoren werden nicht durch Dummy-Vektoren ersetzt. Alle verfügbaren Vektoren werden insgesamt viermal unter verschiedenen Bedingungen durchlaufen. Die Bedingungen sind

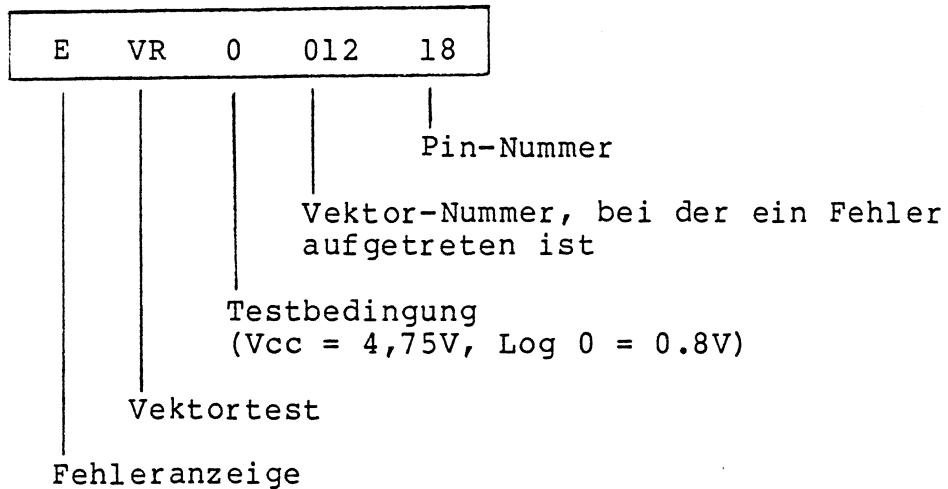
0	Vcc = 4.75 V	Log 0 = 0.8 V
1	Vcc = 5.25 V	Log 0 = 0.8 V
H	Vcc = 4.75 V	Log 1 = 2.4 V
L	Vcc = 5.25 V	Log 1 = 2.4 V

Taste	Anzeige	Bedeutung
	READY MMI 10H8	Das PAL MMI 10H8 ist selektiert.
CHK	CH 00	Check-Art eingeben.
9	CH 09	Check 9 (Vektor-Test) ist gewählt.
ENTER	H CH P> L xxx 0 1	Das zu testende PAL muß spätestens jetzt eingesetzt werden. Die Anzahl der gespeicherten Testvektoren wird angezeigt. Sind keine Vektoren vorhanden, erscheint in der Anzeige CH P> 00 und der Programmierer antwortet mit einem akustischen Signal.
	VR PASS	Vektortest wurde fehlerfrei durchgeführt.
ENTER	READY MMI 10H8	

**Fehlermeldungen** beim Vektor-Test werden standardmäßig mit

VR FAIL

angezeigt, oder, bei Aktivierung von FC 30, in detaillierter Form mit Angabe der fehlerhaften Vektor- und Pin-Nummer. Eine detaillierte Fehlermeldung kann folgendermaßen aussehen:



Diese Fehlermeldung zeigt an, daß im 12. Vektor bei Vcc 4.75 V und Schwelle 0.8 V ein fehlerhafter Zustand an Pin 18 erkannt wurde. Der Vektortest wird nicht weitergeführt, das PAL ist zum Zeitpunkt der Fehlermeldung bereits stromlos.

#### 4.4 FC - Funktionen

Mit der FC-Taste (EPP-80) bzw. Hex-Taste '8' (MPP-80) werden Sonderfunktionen aktiviert, die den Betriebszustand des Programmers ändern. Zusätzlich zu den im EPP-80/MPP-80S Bedienerhandbuch beschriebenen Funktionen sind mit dem UPM-Modul folgende Funktionen wählbar:

FC	30	Detaillierte Fehlermeldung Report ON/OFF
FC	32	Last Fuse-Programmierung ON/OFF

#### FC 30 - Detaillierte Fehlermeldung ON/OFF

Mit der Funktion FC 30 kann die im PAL Mode standardmäßig ausgegebene Fehleranzeige pass/fail in eine genau aufgeschlüsselte Fehleranzeige umgeschaltet werden. Die Meldung gibt dann die fehlerhafte Adresse bzw. Vektor- und Pin-Nummer an. Gleichzeitig mit FC 30 muß jedoch auch FC E0 (Fehlermeldung ON) aktiviert sein.

# BEDIENUNGSANLEITUNG UPM

Taste	Anzeige	Bedeutung
	READY MMI 16R8	Das PAL MMI 16R8 ist selektiert.
FC	FC 00	Wahl des gewünschten FC-Codes.
30	FC 30	Funktion 30 ist angewählt.
ENTER	REPORT OFF	Die aktuelle Einstellung wird angezeigt, hier: REPORT OFF. Mit ENTER wird die Einstellung bestätigt oder mit der '-' Taste in eine detaillierte Anzeige umgeschaltet.
-	REPORT ON	Ausführliche Fehlermeldung wurde gewählt.
ENTER	READY MMI 16R8	Bei allen Fehlermeldungen wird nun statt der Standardmeldung pass/fail die entsprechende Adresse und, soweit sinnvoll, RAM- und PAL-Daten, angezeigt.

## FC 32 - Last Fuse - Programmierung ON/OFF

Mit dieser Funktion kann der Zustand der Security Fuse-Programmierung abgefragt bzw. geändert werden.

Taste	Anzeige	Bedeutung
	READY MMI 16R6	Das PAL MMI 16R6 ist selektiert.
FC	FC 00	Wahl des gewünschten FC-Codes.
32	FC 32	Funktion 32 ist angewählt.
ENTER	LAST FUSE OFF	Die aktuelle Einstellung wird angezeigt, hier: LAST FUSE OFF. Mit ENTER wird die Einstellung bestätigt oder mit der '-' Taste umgeschaltet.
-	LAST FUSE ON	Last Fuse-Programmierung wurde gewählt.
ENTER	READY MMI 16R6	Nach jedem Programmiervorgang fragt der Programmierer nun, ob die letzte(n) Sicherung(en) programmiert werden soll(en).

## 4.5 Ausgabe der UPM-internen Bauteileliste

Das UPM hat eine Liste aller programmierbaren Bauteile eingebaut. die abgerufen und auf einem angeschlossenen Bildschirm bzw. Drucker ausgegeben werden kann.

Die Liste ist aufgeteilt in EPROMs, PALs und BIPOLAR Bauteile und wird jeweils vom entsprechenden Modus aus gestartet.

Taste	Anzeige	Bedeutung
	READY MMI XXX	Ausgangsstellung für Ausgabe der PAL-Liste
OUT	OUT A or B	Wahl des benötigten Kanals durch Eingabe von A oder B über die Hex-Tastatur (nur EPP-80 und wenn 2 Schnittstellen installiert sind).
A	OUT 00	Programmer erwartet Eingabe des gewünschten Codes.
99	OUT 99	Code 99 aktiviert die Ausgabe der entsprechenden Bauteileliste an der selektierten Schnittstelle mit der selektierten Baudrate.
ENTER	=====	Beginn der Übertragung

Die Ausgabe der Liste kann am Terminal mit CNTRL S angehalten und mit CNTRL Q wieder fortgesetzt werden.

## 5. DATEN EIN-/AUSGABE

Für die Datenübertragung sind folgende Übertragungsprotokolle verfügbar:

Code	Protokoll
-----	-----
IN 90	JEDEC
IN 91	MMI HEX (*)
IN 40	INTEL HEX (*)
OUT 90	JEDEC
OUT 91	MMI HEX (*)
OUT 92	FUSE PLOT
OUT 40	INTEL HEX (*)
-----	-----

Für Beispiele beziehen Sie sich bitte auf Kapitel 5.2.

(\*) nicht für Bausteine mit 'Polarity fuses' und 'product term sharing'.

### 5.1 JEDEC Datenübertragungsprotokoll für PALs

Dieses Protokoll erlaubt eine genormte Übertragung von Programmierinformation zwischen einem Logikassembler und einem Programmiergerät. Der Unterschied zu anderen Protokollen (z.B. MMI HEX) besteht darin, daß hier wichtige Zusatzinformationen mit übertragen werden können. Dies sind unter anderem Testvektoren, Prüfsummen, PAL-spezifische Parameter etc.

Im JEDEC-Protokoll werden die Programmierinformationen nicht direkt als Speicherplatzinhalte dargestellt, sondern nach speziellen Richtlinien formatiert und übertragen.

#### **Aufbau des JEDEC-Protokolls**

- 1. Beginn der Übertragung:** <stx> - Zeichen (= 02 Hex)  
bis zu einem \* (Asterisk)

Nach diesem Zeichen können beliebige Zeichen übertragen werden (z.B. Datum, Kommentare ...). Der Empfänger startet nach Erhalt des <stx>-Zeichens eine Checksummenberechnung, die jedes nun folgende Zeichen mitprotokolliert. Den Abschluß dieses wie aller Felder bildet der Asterisk (\*).

Syntax:        <STX> ... Kommentare etc ... \* <CR> <LF>

## 2. Device Code

Dieser Teil teilt dem Empfänger den Bausteintyp mit. Das Programmiergerät kann somit alle für diesen Typ erforderlichen Parameter einstellen.

Dieser Code ist dem KONTRON 'Module Selecting Guide' zu entnehmen.

Syntax: D2217\* <CR> <LF>

D: Start des Identifier-Feldes  
22: Herstellercode  
17: Typenidentifikation  
\*: Abschluß des Feldes

Institut für Informatik  
Abt. Computer - Systeme  
Hardware - Labor  
Breitwiesenstr. 20 - 22  
7000 Stuttgart 80

Falls dieses Feld nicht übertragen wird, so wird die Meldung E IN 2 ausgegeben. Nach Betätigen von ENTER kann die Übertragung fortgesetzt werden. Das Programmiergerät wird die empfangenen Daten jetzt nur entsprechend dem momentan eingestellten Typ verteilen. Das heißt, es können keine logischen Fehler mehr erkannt werden. Es können nur Fehler wie z.B. "Zu viele Sicherungen übertragen" erkannt werden.

## 3. Linkinformation

Hier wird die Linknummer nebst Linkinformation übertragen. Nicht vorhandene Ziffern werden als Nullen interpretiert. Im Linkfeld vorhandene Leerzeichen und <CR> <LF>-Zeichen werden nicht berücksichtigt. Falls sich eingegebene Informationen überschneiden, gilt stets das zuletzt angegebene.

Eine mögliche Zeile Linkinformation könnte z.B. folgendermaßen aussehen:

Syntax:

L0000 0101001100 1010100110 0001110111 <CR> <LF>  
L0040 0011110010 1011100111 0110000101\*

Lxxx: Linkernummernidentifikation mit 4-stelliger Linkernummer.  
Blanks und <CR> <LF> in der Linkzeile optional

\*: Abschluß der Linkzeile

Anmerkung: Folgt nach einer Zeile mit <CR> <LF> kein Asterisk, so bleibt der zuletzt angegebene Feldname erhalten.

## 4. Checksummenfeld

Hier kann eine Fuse-RAM-Checksumme mit dem JEDEC-Protokoll übertragen werden. Jedoch ist diese Checksumme abhängig von der Organisation der Felder im Datenspeicher. Für Vergleichszwecke wäre also Voraussetzung, daß alle Programmiergeräte dieselbe Datenspeicherorganisation besitzen.

## 5. Defaultfeld

Hier kann der Datenspeicher komplett mit Nullen oder Einsen gefüllt werden. Dies ist nur sinnvoll bei PAL's mit voll programmierbarem Array, wo man aus irgendwelchen Gründen nur die Zeilen übertragen will, die Programmierinformationen enthalten.

Syntax:        F0\*        Fülle mit Nullen  
                  F1\*        Fülle mit Einsen

## 6. Testvektoren

In diesem Feld können Testvektoren übertragen werden, die sequentiell an das Bauteil übertragen werden. Die Reihenfolge ist von links nach rechts, sofern im P-Feld nicht ein anderes Pinout übertragen wurde.

Syntax:        V0000 0110 0110 ON10 1101 110N\*

V0000:        Start Vektor mit 4-stelliger Nummer  
                  Leerzeichen haben keinen Einfluß

Da bei der Anwendung von Testvektoren mehr Informationen als im Linkfeld benötigt werden, sind neben Nullen und Einsen noch folgende Zeichen gültig:

0 ---	Eingang auf log. low bringen
1 ---	Eingang auf log. high bringen
2-9 ---	Eingang auf > ttl bringen
C ---	Eingang mit Takt beschalten (low, high, low)
F ---	Ein(Ausgang) in Tristate schalten
H ---	Ausgang auf log. high testen
K ---	Eingang mit Takt beschalten (high, low, high)
L ---	Ausgang auf log. low testen
N ---	Versorgungs- und nicht getestete Pins
P ---	Setze Register
X ---	Nicht zu testender Ausgang, Eingang ist offen
Z ---	Eingang oder Ausgang auf tristate testen

## 7. Security Fuse Feld

Hier kann dem Programmiergerät mitgeteilt werden, ob nach erfolgreichem Programmieren und Arrayvergleich die letzte(n) Sicherungen programmiert werden sollen. Diese Sicherungen verhindern, daß das Programmierarray dieses Bausteins wieder gelesen werden kann. Ist dies der Fall, ist der Baustein zwar voll funktionsfähig, es kann jedoch keine funktionsgleiche Kopie davon hergestellt werden. Der Vektortest funktioniert unabhängig vom Zustand der letzten Sicherung(en).

Syntax:        G1\*        Sicherungen programmiert  
                  G0\*        Sicherungen intakt (Urzustand)

## 8. Das QF und QP Feld

Hier kann man dem Empfänger mitteilen, wieviele Pins und wieviele programmierbare Sicherungen der Baustein enthält. Dies ist für ein intelligentes Programmiergerät jedoch nicht von Belang, da hier keine Information über den Baustein selbst enthalten ist. Der Aufbau des Chips wird im Definitionsfeld übergeben. Man kann diese Information höchstens als Zusatzinformation betrachten, um den im Device Code übertragenen Typ zu vergleichen.

Syntax:        QF 267\*        267 programmierbare Sicherungen  
                  QP 24\*        24 Pin-Baustein

## 9. Kommentarfeld

Hier können für das Programmiergerät nicht notwendige Informationen übertragen werden. Es kann bis auf den Asterisk alle druckbaren Zeichen enthalten.

Syntax:        N            Dies ist eine Kommentarzeile \*

## 10. Ende der Übertragung

Die Übertragung wird vom Programmiergerät beim Empfang eines <etx> (03 Hex) abgebrochen.

## 11. Übertragungsschecksumme

Nach dem Ende der Übertragung folgen noch vier Zeichen. Das ist die Checksumme über alle übertragenen Zeichen, einschließlich <stx> und <etx>. Das Programmiergerät bildet ebenfalls von allen empfangenen Zeichen eine Checksumme und vergleicht beide. Ungleichheit läßt auf fehlerhafte Übertragung schließen. Beim Empfang von 4 Nullen werden die Checksummen nicht verglichen.

## 5.2 Datenformat - Beispiele

On the following pages the example from the PAL module is presented in variation

1. for the PALASM

2. for CUPL

including the corresponding output.

This is the source file for the PALASM.

PAL12H6  
PAT1.1  
PALNAME

PAL DESIGN SPECIFICATION  
(COMMENTARY)  
08.10.84

C D F G M N P Q I GRD  
J K L R O H E B A VCC

$B = \neg A$

$E = C * D$

$H = F + G$

$O = \neg M * \neg N$

$R = P * \neg Q + \neg P * Q$

$L = \neg I + \neg J + \neg K$

DESCRIPTION

The fuse plot of the PALASM

PALNAME

08.10.84

	0123	4567	8901	2345	6789	0123	4567	8901	
			11	1111	1111	2222	2222	2233	
0	0000	0000	0000	0000	0000	0000	0000	0000	
1	0000	0000	0000	0000	0000	0000	0000	0000	
2	0000	0000	0000	0000	0000	0000	0000	0000	
3	0000	0000	0000	0000	0000	0000	0000	0000	
4	0000	0000	0000	0000	0000	0000	0000	0000	
5	0000	0000	0000	0000	0000	0000	0000	0000	
6	0000	0000	0000	0000	0000	0000	0000	0000	
7	0000	0000	0000	0000	0000	0000	0000	0000	
8	----	---X	--00	--00	--00	--00	----	----	/A
9	XXXX	XXXX	XX00	XX00	XX00	XX00	XXXX	XXXX	
10	XXXX	XXXX	XX00	XX00	XX00	XX00	XXXX	XXXX	
11	XXXX	XXXX	XX00	XX00	XX00	XX00	XXXX	XXXX	
12	0000	0000	0000	0000	0000	0000	0000	0000	
13	0000	0000	0000	0000	0000	0000	0000	0000	
14	0000	0000	0000	0000	0000	0000	0000	0000	
15	0000	0000	0000	0000	0000	0000	0000	0000	
16	X-X-	----	--00	--00	--00	--00	----	----	C*D
17	XXXX	XXXX	XX00	XX00	XX00	XX00	XXXX	XXXX	
18	0000	0000	0000	0000	0000	0000	0000	0000	
19	0000	0000	0000	0000	0000	0000	0000	0000	
20	0000	0000	0000	0000	0000	0000	0000	0000	
21	0000	0000	0000	0000	0000	0000	0000	0000	
22	0000	0000	0000	0000	0000	0000	0000	0000	
23	0000	0000	0000	0000	0000	0000	0000	0000	
24	----	X----	--00	--00	--00	--00	----	----	F
25	----	----	X-00	--00	--00	--00	----	----	G
26	0000	0000	0000	0000	0000	0000	0000	0000	
27	0000	0000	0000	0000	0000	0000	0000	0000	
28	0000	0000	0000	0000	0000	0000	0000	0000	
29	0000	0000	0000	0000	0000	0000	0000	0000	
30	0000	0000	0000	0000	0000	0000	0000	0000	
31	0000	0000	0000	0000	0000	0000	0000	0000	
32	----	----	--00	-X00	-X00	--00	----	----	/M*/N
33	XXXX	XXXX	XX00	XX00	XX00	XX00	XXXX	XXXX	
34	0000	0000	0000	0000	0000	0000	0000	0000	
35	0000	0000	0000	0000	0000	0000	0000	0000	
36	0000	0000	0000	0000	0000	0000	0000	0000	
37	0000	0000	0000	0000	0000	0000	0000	0000	
38	0000	0000	0000	0000	0000	0000	0000	0000	
39	0000	0000	0000	0000	0000	0000	0000	0000	

# Bedienungsanleitung UPM

```

40 ---- ---- --00 --00 --00 X-00 -X-- ---- P*/Q
41 ---- ---- --00 --00 --00 -X00 X--- ---- /P*Q
42 0000 0000 0000 0000 0000 0000 0000 0000
43 0000 0000 0000 0000 0000 0000 0000 0000
44 0000 0000 0000 0000 0000 0000 0000 0000
45 0000 0000 0000 0000 0000 0000 0000 0000
46 0000 0000 0000 0000 0000 0000 0000 0000
47 0000 0000 0000 0000 0000 0000 0000 0000

48 ---- ---- --00 --00 --00 --00 ---- -X-- /I
49 ---- ---- --00 --00 --00 --00 ---- ---X /J
50 ---- ---- --00 --00 --00 --00 ---X ---- /K
51 XXXX XXXX XX00 XX00 XX00 XX00 XXXX XXXX
52 0000 0000 0000 0000 0000 0000 0000 0000
53 0000 0000 0000 0000 0000 0000 0000 0000
54 0000 0000 0000 0000 0000 0000 0000 0000
55 0000 0000 0000 0000 0000 0000 0000 0000

56 0000 0000 0000 0000 0000 0000 0000 0000
57 0000 0000 0000 0000 0000 0000 0000 0000
58 0000 0000 0000 0000 0000 0000 0000 0000
59 0000 0000 0000 0000 0000 0000 0000 0000
60 0000 0000 0000 0000 0000 0000 0000 0000
61 0000 0000 0000 0000 0000 0000 0000 0000
62 0000 0000 0000 0000 0000 0000 0000 0000
63 0000 0000 0000 0000 0000 0000 0000 0000

```

LEGEND: X: FUSE NOT BLOWN (L,N,0) -: FUSE BLOWN (H,P,1)  
 0: PHANTOM FUSE (L,N,0) 0: PHAN. FUSE (H,P,1)  
 NUMBER OF FUSES BLOWN = 306

# Bedienungsanleitung UPM

The HEX file of the PALASM.

[illegible]

# Bedienungsanleitung UPM

This is the source file for CUPL.

```

PARTNO      00001 ;
NAME        DEMO  ;
DATE        10/08/84 ;
REV         1 ;
DESIGNER    KLEINHENZ ;
COMPANY     KONTRON MESSTECHNIK ;

/*****
/* This is the example from the manual for PAL-MOD
/* converted to the CUPL language
/*
*****/
/* Allowable Target Device Types: 12H6
*****/

/** Inputs **/

PIN 1 = C ; /*
PIN 2 = D ; /*
PIN 3 = F ; /*
PIN 4 = G ; /*
PIN 5 = M ; /*
PIN 6 = N ; /*
PIN 7 = P ; /*
PIN 8 = Q ; /*
PIN 9 = I ; /*
PIN 11 = J ; /*
PIN 12 = K ; /*
PIN 19 = A ; /*

/** Outputs **/

PIN 13 = L ; /*
PIN 14 = R ; /*
PIN 15 = O ; /*
PIN 16 = H ; /*
PIN 17 = E ; /*
PIN 18 = B ; /*

/** Declarations and Intermediate Variable Definitions **/

/** Logic Equations **/

B = !A ; /* inverter
E = C & D ; /* 2 input AND gate
H = F # G ; /* 2 input OR gate
O = !M # !N ; /* 2 input NOR gate
R = P & !Q # !P & Q ; /* 2 input XOR gate
L = !I # !J # !K ; /* 3 input NAND gate

```

# Bedienungsanleitung UPM

CUPL Version 1.01b Copyright (c) 1983 Assisted Technology, Inc.  
-- LISTING

Source File: D:TEST.pld

Device File: M12H6.pt

```

$0001$          PARTNO      00001 ;
$0002$          NAME        DEMO ;
$0003$          DATE        10/08/84 ;
$0004$          REV         1 ;
$0005$          DESIGNER    KLEINHENZ ;
$0006$          COMPANY     KONTRON MESSTECHNIK ;
$0007$
$0008$  /*****
$0009$  /* This is the example from the manual for PAL-MOD
$0010$  /* converted to the CUPL language
$0011$  /*
$0012$  /*****
$0013$  /* Allowable Target Device Types: 12H6
$0014$  /*****
$0015$
$0016$  /** Inputs **/
$0017$
$0018$  PIN      1      =      C      ;      /*
$0019$  PIN      2      =      D      ;      /*
$0020$  PIN      3      =      F      ;      /*
$0021$  PIN      4      =      G      ;      /*
$0022$  PIN      5      =      M      ;      /*
$0023$  PIN      6      =      N      ;      /*
$0024$  PIN      7      =      P      ;      /*
$0025$  PIN      8      =      Q      ;      /*
$0026$  PIN      9      =      I      ;      /*
$0027$  PIN     11      =      J      ;      /*
$0028$  PIN     12      =      K      ;      /*
$0029$  PIN     19      =      A      ;      /*
$0030$
$0031$  /** Outputs **/
$0032$
$0033$  PIN     13      =      L      ;      /*
$0034$  PIN     14      =      R      ;      /*
$0035$  PIN     15      =      O      ;      /*
$0036$  PIN     16      =      H      ;      /*
$0037$  PIN     17      =      E      ;      /*
$0038$  PIN     18      =      B      ;      /*
$0039$
$0040$  /** Declarations and Intermediate Variable Definitions **/
$0041$
$0042$  /** Logic Equations **/
$0043$
$0044$  B = !A          ;      /* inverter
$0045$  E = C & D        ;      /* 2 input AND gate
$0046$  H = F # G        ;      /* 2 input OR gate
$0047$  O = !M # !N      ;      /* 2 input NOR gate
$0048$  R = P & !Q # !P & Q ;      /* 2 input XOR gate
$0049$  L = !I # !J # !K ;      /* 3 input NAND gate
$0050$

```

JEDEC sum check (1EB6)

# Bedienungsanleitung UPM

CUPL 1.01b Copyright (c) 1983 Assisted Technology, Inc.  
 -- FUSE TABLE

Source File: D:TEST.pld

Device File: M12H6.pt

partno 00001  
 name DEMO  
 rev 1  
 date 10/08/84  
 designer KLEINHENZ  
 company KONTRON MESSTECHNIK

column	0123	4567	8901	11	1111	1111	2222	2222	2233	
				2345	6789	0123	4567	8901		
row 8	----	---x	--	--	--	--	----	----	!A	
row 16	x-x-	----	--	--	--	--	----	----	D&C	
row 24	----	----	x-	--	--	--	----	----	G	
row 25	----	x---	--	--	--	--	----	----	F	
row 32	----	----	--	--	-x	--	----	----	!N	
row 33	----	----	--	-x	--	--	----	----	!M	
row 40	----	----	--	--	--	-x	x---	----	!P&Q	
row 41	----	----	--	--	--	x-	-x--	----	P&!Q	
row 48	----	----	--	--	--	--	---x	----	!K	
row 49	----	----	--	--	--	--	-----	---x	!J	
row 50	----	----	--	--	--	--	-----	-x--	!I	

pins	2211	3399	4400	5500	6600	7700	8822	9911	
polarity	HLHL	HLHL	HLHH	HLHH	HLHH	HLHH	HLHL	HLHL	

pin: 1 C  
 pin: 2 D  
 pin: 3 F  
 pin: 4 G  
 pin: 5 M  
 pin: 6 N  
 pin: 7 P  
 pin: 8 Q  
 pin: 9 I  
 pin: 11 J  
 pin: 12 K  
 pin: 13 L  
 pin: 14 R  
 pin: 15 O  
 pin: 16 H  
 pin: 17 E  
 pin: 18 B  
 pin: 19 A

# Bedienungsanleitung UPM

The HEX file in MMI format from CUPL

[illegible]

\$SOE40

The JEDEC output file of CUPL.

00001  
DEMO  
1  
10/08/84  
KLEINHEINZ  
KONTRON MESSTECHNIK

\*D2219

\*QP20

\*OF0384

\*FO

\*G0

```
*L0000 11111110111111111111111111111111
```

\*L0024 0000000000000000000000000000

\*L0048 00000000000000000000000000000000

\*L0072 0000000000000000000000000000000000

```
*L0096 01011111111111111111111111111111
```

\*L0120 00000000000000000000000000000000

```
*L0144 111111110111111111111111
```

```
*L0168 111110111111111111111111111111
```

```
*L0192 11111111111111011111111111
```

```
*L0216 111111111110111111111111
```

```
*L0240 11111111111111110011111111
```

```
*L0264 11111111111111110110111111
```

```
*L0288 11111111111111111111111111111111
```

```
*L0312 11111111111111111111111111111110
```

```
*L0336 1111111111111111111111111111011
```

```
*L0360 00000000000000000000000000000000
```

\*C1EB6

\*7574

## 6. UPM FEHLERMELDUNGEN

Das UPM zeichnet sich durch eine umfangreiche Palette an Bauteile- und Modultests aus. Zusätzlich zu den im EPP-80/MPP-80S Bedienerhandbuch beschriebenen Fehlermeldungen können folgende Anzeigen erscheinen:

<u>Anzeige</u>	<u>Bemerkungen</u>
ERROR ADDR	Kurzschluß auf Adressleitung
ERROR DATA	Kurzschluß auf Datenleitung
ERROR REVERSE	Bauteil verkehrt eingesetzt
ERROR CURRENT	Kurzschluß auf Vcc
ERROR SOCKET	Bauteil befindet sich im falschen Sockel
ERROR NO DEV	Es befindet sich kein Bauteil im Sockel
ERROR PIN XX	Pin XX ist offen oder defekt
ERROR LA FUSE	Sicherung für Kopierschutz des im Sockel befindlichen PALs wurde bereits programmiert
ERROR PATTERN	Phantom Pattern stimmt nicht mit dem selektierten PAL überein
ERROR SYSTEM	Gewählter Baustein wird nicht unterstützt oder es wurde versucht, im Bipolar- oder PAL-Mode bei Funktion FC El (Fehlermeldungen OFF) ein Bauteil zu programmieren.
ERROR CAL XXX	Spannungskalibrierung außer Toleranz
ERROR HARD TEST XXX	Hardware-Fehler (Hybrid/Gate Array etc.)
ERROR SILICON	Silicon Signature stimmt nicht mit gewähltem Baustein überein

### Fehlermeldungen bei der MOVE/INSert-Funktion)

E SAVE 0	Unbekanntes oder ungeeignetes Bauteil wurde als Backup-Medium eingesetzt (Bauteile mit Silicon Signature müssen von Intel sein)
E SAVE 1	Die gewählte EPROM-Größe ist zu klein für den PAL-Inhalt mit Testvektoren
E SAVE 2	Das für den Backup verwendete EPROM ist nicht leer
E SAVE 3	Das Bauteil ist nicht programmierbar (evtl. falscher Typ selektiert)

## Bedienungsanleitung UPM

### UPM-Fehlermeldungen

E	LOAD	0	Unbekanntes EPROM im Sockel
E	LOAD	1	Prüfsumme der im EPROM gespeicherten Daten ist falsch, oder das Bauteil enthält keine vorher mit MOV abgespeicherten Daten.
E	LOAD	2	Daten von einem nicht implementierten PAL sollen geladen werden.

### Fehlermeldungen bei Datenübertragungen

E	IN	0	Falsche Checksumme übertragen (mit ENTER kann die Funktion fortgesetzt, mit '-' abgebrochen werden)
E	IN	1	Unbekannter Bauteil-Code (Funktion wird abgebrochen)
E	IN	2	Kein Bauteil-Code übertragen. Der UPM-Select-Code wird im Display angezeigt. Stimmt er mit dem gewünschten Pal-Typ überein, kann die Funktion mit ENTER fortgesetzt werden; andernfalls kann der 4-stellige UPM-Select-Code des gewünschten PALs mit dem Hex-Tastenfeld eingegeben werden. Mit '-' wird die Funktion abgebrochen.
E	IN	3	Anzahl der übertragenen Sicherungen ist zu groß (Funktion wird abgebrochen).
E	IN	4	RAM-Bereich ist zu klein für JEDEC Dateneingabe (Funktion wird abgebrochen).
E	IN	5	Ungültige Zeichen übertragen beim MMI Hex Format.
E	IN	6	Die Anzahl der Komponenten der übertragenen Testvektoren ist zu groß oder zu klein für den eingestellten Pal-Typ.

## 7. CHECKSUMMEN ALGORITHMEN

Die KONTRON Programmiergeräte EPP-80/MPP-80S ermöglichen es dem Anwender, eine Checksumme von einem PROM oder von dem Anwender-RAM zu erzeugen. Eine Checksumme ist eine bestimmte Zahl (EPP-80/MPP-80 verwenden eine vierstellige Anzeige für die Checksumme) und ist von dem Inhalt und der Größe des PROM's abhängig. EPP-80/MPP80S bieten zwei verschiedene Arten von Checksummen:

**Additive Checksumme**  
**KONTRON CRC-Checksumme**

### 7.1 Additive Checksumme

Diese Checksumme basiert auf der Addition sämtlicher Datenbytes eines vom Anwender definierten Datenblockes (entweder von einem eingelegten PROM oder von einem gewünschten Bereich des Anwender-RAMs). Für diese Checksumme wird die Reihenfolge bei der Addition der Bytes nicht berücksichtigt. Das Resultat bleibt gleich, ob die Checksumme von der Start- bis zur Endadresse oder von der End- bis zur Startadresse erzeugt wird. Außerdem erkennt diese Checksumme keinen Unterschied, wenn ein Bit an einer Stelle von Low auf High und an einer anderen Stelle das gleiche Bit von High auf Low geht.

Die folgenden Zeilen zeigen ein Beispiel für die Erzeugung einer vierstelligen Checksumme aus drei Datenbytes:

addr	data
<hr style="border: none; border-top: 1px dashed black;"/>	
0000	C3
0001	55
0002	AA
<hr style="border: none; border-top: 1px dashed black;"/>	
01C2 = dargestellte Checksumme	

Diese Checksumme wird automatisch nach jedem Check 3 (eine Checksumme von einem eingesetzten PROM) oder nach jedem Check 4 (eine Checksumme von einem gewünschten Anwender-RAM-Bereich) dargestellt und besteht aus einer vierstelligen Zahl.

Die Nachteile der additiven Checksumme sind aus dem nachfolgenden Beispiel ersichtlich:

addr	hex	binär		hex	binär	
<u>0000</u>	<u>C3</u>	<u>1100</u>	<u>0011</u>	<u>C1</u>	<u>1100</u>	<u>0001</u>
0001	55	0101	0101	57	0101	0111
0002	AA	1010	1010	AA	1010	1010
Additive						
Checksumme	01C2			01C2		

Die Checksummen bleiben gleich, obwohl Bit 1 im Byte 0 und 1 unterschiedliche Werte aufweist.

## 7.2 KONTRON CRC-Checksumme

Bei dieser Art von Checksumme sind die Nachteile der additiven Checksumme nicht vorhanden. Deshalb entsteht bei geänderter Reihenfolge der 'addierten' Bytes oder bei zwei geänderten Bits eine andere Checksumme (obwohl das Ergebnis bei der Additiven Checksumme das gleiche wäre).

Wenn die Checksumme eines PROM's auf das PROM geschrieben wird ist es möglich, die Daten in dem gesamten PROM zu überprüfen, lediglich durch Vergleichen der geschriebenen Checksumme mit der erzeugten Checksumme. Dies ist ein sehr nützliches Werkzeug bei der Fehlersuche oder für den Techniker in Servicefällen. Die KONTRON Checksummenmethode verwendet einen Algorithmus, der dem CRC-Check in Floppy-Laufwerken ähnelt.

### Erzeugung der Kontron CRC Checksumme

Die KONTRON Checksumme wird Bit-für-Bit erzeugt. Wenn die Checksumme aus einem 4-Bit-PROM erzeugt werden soll, wird das obere Nibble (Bit 4-7) auf Low gelegt. Zunächst muß ein 16-Bit-Puffer mit 0 initialisiert werden (nachdem die Erzeugung der Checksumme abgeschlossen ist, wird dieser Puffer die Checksumme enthalten). Der nachfolgende Abschnitt beschreibt die Erzeugung eines einzelnen Bits:

Das niedrigstwertige Bit (LSB) des Datenbytes - das erste Bit bei der Erzeugung einer Checksumme - wird mit Bit 1 des Puffers durch exklusives logisches ODER verbunden. Das Ergebnis des exklusiven ODERs wird exklusiv mit Bit 3 des Puffers verodert; dieses Ergebnis wird mit Bit 5 des Puffers exklusiv verodert; und schließlich wird dieses Ergebnis mit Bit 14 des Puffers wieder exklusiv verodert. Dann wird der Inhalt des Puffers mit 2 multipliziert und das Ergebnis des exklusiven ODERs wird in Bit 0 des Puffers (LSB) einkopiert.

Dieser Vorgang ist siebenmal zu wiederholen, anfangend mit Bit 7 (MSB des Datenbytes) bis Bit 1 des Datenbytes. Nachdem 1 Byte in der Checksummenerzeugung verarbeitet wurde, enthält der Puffer selbst 1 Byte. Das obere Byte des Puffers enthält noch 0. Die folgenden Zeilen zeigen ein Beispiel der Erzeugung einer KONTRON CRC-Checksumme über 3 Bytes. Die Datenbytes sind die gleichen die im vorigen Beispiel verwendet wurden:

addr	data	bit	result	Checksumme				hex
				binary				
0000	C3	0	1	0000	0000	0000	0001	0001
		7	1	0000	0000	0000	0011	0003
		6	0	0000	0000	0000	0110	0006
		5	1	0000	0000	0000	1101	000D
		4	1	0000	0000	0001	1011	001B
		3	0	0000	0000	0011	0110	0036
		2	0	0000	0000	0110	1100	006C
		1	1	0000	0000	1101	1001	00D9

# Bedienungsanleitung UPM

addr	data	bit	result	binary				hex
0001	55	0	0	0000	0001	1011	0010	01B2
		7	0	0000	0011	0110	0100	0364
		6	0	0000	0110	1100	1000	06C8
		5	1	0000	1101	1001	0001	0D91
		4	1	0001	1011	0010	0011	1B23
		3	0	0011	0110	0100	0110	3616
		2	0	0110	1100	1000	1100	6C8C
		1	0	1101	1001	0001	1000	D918
0002	AA	0	0	1011	0010	0011	0000	B230
		7	0	0110	0100	0110	0000	6460
		6	0	1100	1000	1100	0000	C8C0
		5	0	1001	0001	1000	0000	9180
		4	0	0010	0011	0000	0000	2300
		3	1	0100	0110	0000	0001	4601
		2	1	1000	1100	0000	0011	8C03
		1	0	0001	1000	0000	0110	1806

Es ergibt sich somit eine Checksumme von 1806. Wenn Sie  
versuchen, die Reihenfolge der Datenbytes zu ändern, werden Sie  
feststellen, daß eine andere Checksumme erzeugt wird.

7.3 Programmierbeispiel für additive und CRC Checksumme

```

;
; z80 source code for checksum
; calculation
;
; checksum calculation
;
; load hl to begin of datamemory
; load bc with length of datafield
; result is in register de on end
;
maksum: ld de,0
        ld (crcbit),de ;reset buffer for result
        ld hl,begin
        ld bc,length
        ;
        ;address prom
        ;
chec32: ld a,(hl)
        ;
        call addsum      ;if additive checksum
        ;
        call crcsum      ;if kontron checksum
        ;
        inc hl           ;next byte
        dec bc           ;! done
        ld a,b
        or c
        jr nz,chec32     ;until end
        ld de,(crcbit)   ;sum to de
        ret
        ;
        ;calculate additive sum
        ;subroutine
        ;
addsum: push de
        ld de,(crcbit)   ;get buffer
        push hl
        ;
        and 00001111b    ;if 4bit prom
        ;
        ld l,a           ;get data
        ld h,0
        add hl,de
        ld (crcbit),hl   ;save result
        pop hl
        pop de
        ret
        ;
        ;subroutine for crc promcheck
        ;polynom  $x^1+x^3+x^5+x^{14}$ 
        ;

```

## Programmierbeispiel für additive und CRC Checksumme (Fortsetzung)

```

chsum:  push bc
        push de
        ld de,(crcbit)  ;get buffer
        ld b,8           ;create 8 bits
        ;
        and 00001111b    ;if 4bit
        ;
        ld c,a           ;save data byte
crcbil: ld a,c
        and 1            ;look for bit 0
        rrc e            ;rotate crcbuffer
        xor e            ;xor bit 0 of data and bit 1 crc
        rrc e
        rrc e
        xor e            ;same for 3 crc buffer
        rrc e
        rrc e
        xor e            ;same for 5 crcbuffer
        rrc e
        rrc e
        rrc e            ;buffer to old position
        ;
        ;next for x14
        ;
        rlc d
        rlc d
        xor d
        rrc d
        rrc d
        rrc a            ;crc bit in carry
        rl e             ;cy in bit 0(e),bit 7(e) in cy
        rl d             ;cy in bit0(d),bit 7(d) in cy
        rlc c            ;next databit to bit 0
        djnz crcbil      ;continue
        ld (crcbit),de   ;save result
        pop de
        pop bc
        ret
        ;
crcbit: ds 2             ;2 ram places for sumbuffer
        ;
        ;end

```

KONTRON  
ELECTRONIC  
GROUP

KONTRON  
ELECTRONIC  
GROUP



KONTRON MESSTECHNIK GmbH  
— Application Dptm. —

Oskar-von Miller-Straße 1  
D-8057 Eching/Munich  
W.-Germany

- ☐ Software Discrepancy Report  
☐ Hardware Discrepancy Report  
☐ Document Discrepancy Report  
☐ Problem      ☐ Suggestion

Date \_\_\_\_\_

From

Name

Function/Organisation

Address

Phone

Please use this form to report errors, bugs or problems encountered with our products. You can also submit your suggestions for enhancements of our hardware, software and manuals. We need your cooperation to guarantee the optimum in system's performance. Please, preferably, fill in in English and use one form per problem.

System, Title of Diskette/Manual

Order No.

Serial No./Rev.

Options/Operating System/Program Name

Brief statement of Problem/Suggestion

DETAILS (Please include listings, diskettes, copies of manual etc. as appropriate)

---

---

---

---

---

---

---

---

---

---

Incl. ☐ Diskette      ☐ Listing      ☐ Other \_\_\_\_\_

## DATE RECEIVED

**ACTION BY:**

☐ PROBLEM IDENTIFIED – FIX TO BE INCLUDED IN RELEASE

☐ PROBLEM FIXED IN RELEASE.

☐ PROBLEM CANNOT BE REPRODUCED

EXPLANATION BY: \_\_\_\_\_

DATE:

☐☐☐☐☐☐ LOG BOOK☐ IMMEDIATE MNGR.

☐ ORIGINATOR

☐ APPS. ENGR./TECHN. PUBL.☐ PROD. MNGR.

Eching, August 1986

```

*****
*
*
*
*
*
*
*
*
*
*
*
*****

```

KUNDENINFORMATION

ÜBER MDM/ UPM - PROGRAMMIERMODUL-SOFTWARE

Ihr MDM MOS Device Modul oder UPM Universal-Programmier- und Testmodul wurde mit der am Tag der Herstellung aktuellen Softwarerevision ausgeliefert. Damit Ihr KONTRON Programmiermodul ständig den neuesten Marktentwicklungen entspricht, werden neu erscheinende Bauteile kontinuierlich in die bestehende Modulsoftware implementiert und zu gegebener Zeit von KONTRON als Software-Update für das jeweilige Programmiermodul veröffentlicht.

Kunden, die diesen Service in Anspruch nehmen möchten, bietet KONTRON einen Modul-Software-Vertrag an. Wenn Sie das anhängende Formblatt ausgefüllt und unterschrieben zurückschicken, erhalten Sie für die Dauer eines Jahres automatisch jede neue Software-Revision die nach Unterzeichnungsdatum für das durch den Vertrag abgedeckte Programmiermodul freigegeben wird. Bei Vertragsunterzeichnung bereits bestehende, neuere Software-revisionen schließt der Vertrag jedoch nicht ein. Diese können nur getrennt bestellt werden.

Als Alternative zu einem Softwarevertrag haben Sie jedoch auch die Möglichkeit, jede neue Modul-Software einzeln zu bestellen. Nähere Auskünfte gibt Ihnen gerne eines unserer technischen Büros.

Jede Softwarelieferung, egal ob im Rahmen eines Vertrages oder als Einzellieferung, besteht aus einem Satz Software-PROMs sowie der dazugehörigen Dokumentation.

Eventuelle Hardware-Updates werden, falls benötigt, gesondert angeboten und in Rechnung gestellt.



SOFTWARE-VERTRAG

FÜR KONTRON PROGRAMMIERMODULE DER SERIE MDM ODER UPM

HIERMIT SCHLIESSE ICH ZU DEN AM TAG DER UNTERZEICHNUNG GÜLTIGEN PREISBEDINGUNGEN EINEN MODUL-SOFTWAREVERTRAG FÜR MEIN MDM/UPM-A/UPM-B/UPM-C\* PROGRAMMIERMODUL MIT DER SERIENNUMMER ..... AB. DER VERTRAG BERECHTIGT MICH ZUM BEZUG ALLER SOFTWARE-UPDATES, DIE KONTRON WÄHREND DER FOLGENDEN 12 MONATE FÜR DAS DURCH DEN VERTRAG ABGEDECKTE MODUL VERÖFFENTLICHT.

DER SOFTWARE-VERTRAG TRITT AM TAG DER UNTERZEICHNUNG IN KRAFT UND HAT EINE FESTE LAUFZEIT VON 1 JAHR. DER VERTRAG VERLÄNGERT SICH AUTOMATISCH UM EIN WEITERES JAHR, WENN ER NICHT 90 TAGE VOR ABLAUF SCHRIFTLICH GEKÜNDIGT WIRD.

DIESER VERTRAG IST NICHT ÜBERTAGBAR UND NUR GÜLTIG FÜR DAS IN DIESEM FORMULAR SPEZIFIZIERTE PROGRAMMIERMODUL.

\* MDM MOS DEVICE MODUL  
UPM UNIVERSAL PROGRAMMIERMODUL VERSION A, B, ODER C

NICHTZUTREFFENDES BITTE STREICHEN

.....  
Datum

.....  
Unterschrift

Bitte senden Sie dieses  
Formular unterschrieben  
zurück an:

Empfängeranschrift für  
Software-Updates:

KONTRON MESSTECHNIK GMBH

Name:.....

ABTLG. PROM PROGRAMMER

Firma:.....

Abtlg.:.....

OSKAR-VON-MILLER-STR.1

Strasse:.....

8057 ECHING

Ort:.....

Tel.:.....

Modultyp:.....

Ser.Nr. ....

