

PEPS-III

Programmierbarer
Eprom-Simulator

Handbuch

 **CONITEC DATENSYSTEME**
64807 Dieburg • Dieselstr. 11c • fon 06071-92520 • fax 06071-925233

PEPS-III Handbuch

Stand: 21. Juli 1994

Copyright: © CONITEC 1989 - 1994

Dieselstr. 11c
64807 Dieburg

**Software: Bernhard Emese
Andreas May**

**Handbuch: Johannes C. Lotter
Andreas May**

Kein Teil dieser Dokumentation darf ohne ausdrückliche schriftliche Genehmigung von CONITEC reproduziert, vervielfältigt, gespeichert oder übersetzt werden. Für die Richtigkeit der hier angegebenen Daten übernehmen wir keine Haftung. Änderungen, die dem Fortschritt dienen, behalten wir uns vor, auch ohne dies besonders anzukündigen.

Inhaltsverzeichnis

| | |
|--|----|
| 1 Einführung | 2 |
| 2 Schnellstart | 4 |
| 3 Die Software | 6 |
| 3.1 Anmerkung zu INTEL-HEX Dateien | 9 |
| 3.2 Anmerkung zu MOTOROLA-S Dateien | 10 |
| 3.3 Anmerkung zu Binärdateien | 10 |
| 3.4 Ausgabe der Daten in eine Datei | 10 |
| 3.5 Patchen | 11 |
| 3.6 Datensicherheit | 11 |
| 3.7 Debug-Modus | 12 |
| 3.8 16- und 32-Bit-Zielsysteme | 12 |
| 4 Steckerbelegung | 14 |
| 5 Die Hardware-Programmierung von PEPS | 16 |
| 5.1 Adresse einstellen | 17 |
| 5.2 Datenbyte senden | 17 |
| 5.3 Datenbyte laden | 18 |
| 5.4 RAM-Simulation | 19 |

1

Einführung

PEPS ist ein Hilfsmittel zum Entwickeln von Steuersoftware ("Firmware") für Micro- bzw. Einplatinencomputer (z.B. CEPAC oder EPAC). Dabei steckt der PEPS (Programmierbarer EPROM-Simulator) im EPROM-Sockel, übernimmt die Funktion des EPROMs und erlaubt ähnlich wie ein CPU-Emulator das schnelle Austesten der Software. Weiterhin können einige RAM-Typen simuliert werden, so daß der Arbeitsspeicher des Micro- bzw. Einplatinencomputers auch in den PC eingelesen werden kann. Es ist zu beachten, daß bei RAM-Simulation das CE-Signal nicht ausgewertet wird.

PEPS wird von einem IBM-kompatiblen PC über die Druckerschnittstelle mit dem auszutestenden Programm geladen. Da nur das EPROM simuliert wird, ist für PEPS der CPU-Typ des Zielsystems ohne Belang. Die Software unterstützt auch Zielsysteme mit 16-Bit- oder 32-Bit-Datenbus.

Technische Daten:

Übertragungsrate bis 32.000 Byte/sec. (auf 386er)

Stromaufnahme (bei Zugriff) kleiner 60 mA

Zugriffszeit kleiner 120 ns

Wie man sieht, sind die technischen Spezifikationen in jedem Falle besser als die des zu simulierenden EPROMs. Die hohe Datenübertragungsrate erspart Zwangspausen; PEPS ist in Sekunden programmiert.

Während der Datenübertragung und während des Testlaufs bleibt PEPS mit dem PC verbunden. Die Daten gehen auch nach Ausschalten des Systems oder Abziehen des Kabels nicht verloren. Die Stromversorgung erfolgt über das Zielsystem; die Stromaufnahme beträgt nur etwa ein Drittel von der des simulierten EPROMs.

PEPS besitzt einen Reset-Ausgang mit einer Prüfklemme. Dieser Ausgang wird während der Datenübertragung auf GND gezogen. Im Simulationsmodus liegt die Prüfklemme hochohmig auf VCC. Die Software unterstützt einen Debug-Modus, in dem über den Reset-Ausgang Impulsketten ausgegeben werden, um das Startverhalten des Zielsystems zu testen (s.u.).

PEPS enthält intern bis zu 512 KByte optional batteriegepuffertes RAM; damit lassen sich EPROM's (2716, 2732, 2764, 27128, 27256, 27512, 27513) und RAM's (6116, 6264, 62256, 511001, 628512) von 2 bis 512 KByte simulieren. Der Speicher-Typ wird über DIL-Schalter eingestellt.

PEPS wird in zwei Versionen angeboten, die von außen an der LED-Farbe zu erkennen sind:

- | | |
|--|-------------|
| 128 KByte Speicher mit Batteriepufferung | ⇒ gelbe LED |
| 512 KByte Speicher mit Batteriepufferung | ⇒ rote LED |

Schnellstart

Damit Sie möglichst schnell mit dem EPROM-Simulator arbeiten können, möchten wir Ihnen an dieser Stelle die wichtigsten Schritte zum Betreiben von PEPS erläutern.

- a) Auf der Diskette finden Sie das Programm PEPS.EXE. Das ist alles, was Sie an Software benötigen. Kopieren Sie es auf Ihre Harddisk oder Ihre Arbeitsdiskette. Sollte sich auf der Diskette zusätzlich eine READ.ME-Datei befinden, so lesen Sie diese; sie enthält neue Informationen, die noch nicht in dieses Handbuch aufgenommen werden konnten.
- b) Stellen Sie mit den Schaltern S1 bis S16 den gewünschten EPROM- bzw. RAM-Typ ein. Die Anzeige der erforderlichen Einstellung der DIL-Schalter ist in der Steuersoftware PEPS.EXE integriert und wird abgerufen, indem man das Programm ohne Dateinamen oder Parameter aufruft:

PEPS

Für ein 64 kB EPROM vom Typ 27512 muß eingestellt werden:

| | | | | | | | | | | | | | | | | | |
|-------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EPROM Größe | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
| 27512 64 kB | X | X | - | X | X | - | - | - | - | - | - | - | - | - | - | X | - |

X: DIL-Schalter On

- d) Stecken Sie das Flachbandkabel in den EPROM-Sockel Ihres Zielsystems (**Polung beachten! PEPS wird bei Falschpolung zerstört!**) und schließen Sie PEPS über das Interface-Kabel an die Parallelschnittstelle LPT1 Ihres PC an (wenn Sie eine andere Parallelschnittstelle benutzen wollen, lesen Sie bitte vorher das Kapitel 'Software'). Wenn Sie einen Baustein simulieren wollen, der weniger als 32 Pins hat, müssen Sie einen Zwischensockel mit der Pinanzahl des Bausteines verwenden. Schalten Sie das Zielsystem ein.

- e) Erstellen Sie eine auf dem Zielsystem lauffähige EPROM-Datei (nehmen wir an, sie habe den Namen **ZIEL.COM**) und starten Sie dann die Übertragung durch den Aufruf von:

PEPS ZIEL.COM

Während der Datenübertragung leuchtet die LED am PEPS auf, die Reset-Klemme liegt auf GND. Wenn die Datenübertragung beendet ist, erlischt die LED und der Reset-Ausgang wird hochohmig. PEPS verhält sich nun wie ein EPROM, das mit **ZIEL.COM** programmiert wurde.

Die Software

Das PEPS-Ladeprogramm dient hauptsächlich dazu, EPROM-Daten in den **Simulator** hinein oder heraus zu transferieren. Darüber hinaus bietet es Möglichkeiten für Dateikonversion, manuelles 'Patching' im simulierten EPROM sowie einen interaktiven Debug-Modus, neben einigen anderen Optionen, die im folgenden beschrieben werden.

Die meisten Compiler, Assembler und Linker erzeugen Code entweder als EXE-File, als Binärdatei oder als ASCII-Datei im INTEL-HEX oder MOTOROLA-S-Format. EXE-Files lassen sich mit dem DOS-Befehl EXE2BIN in Binärdateien umwandeln; die anderen Formate werden vom PEPS-Lader erkannt, binär umgewandelt und übertragen. Der Transfer erfolgt mit einer (rechnerabhängigen) Geschwindigkeit von ca. 10 - 32 KByte/sec.

Die Ausgabe erfolgt wahlweise über einen der Parallel-Ports LPT1 bis LPT4. Für Zielsysteme mit 16-Bit oder 32-Bit Datenbus können mit dem Lader Daten auch an 2 bzw. 4 Pepsgeräte gleichzeitig über nur eine Parallel-Schnittstelle übertragen werden. Daneben existiert auch die Möglichkeit, die Daten statt zum PEPS-Simulator in eine Datei mit identischem Inhalt zu schreiben.

Fehlermeldungen (z.b. falsches Dateiformat, Prüfsummenfehler oder Übertragungsfehler) sowie Kontrollmeldungen wie Anzahl der übertragenen Bytes und Dateiformat werden auf dem Bildschirm ausgegeben.

Aufruf:

PEPS [?] gibt eine Liste der verfügbaren Optionen sowie die DIL-Schaltertabelle aus.

PEPS [**<Name>**] [Optionen] überträgt Daten zum PEPS.

<Name>: Pfadname der Datei mit EPROM-Daten, die geladen werden soll (Quelldatei, **muß** vor den Optionen angegeben werden).

Optionen: werden durch ein Minuszeichen '-' eingeleitet, gefolgt von dem Optionsbuchstaben und ggfs. einem Parameter, meist einer Zahl oder einem Dateinamen.

Wichtig: Zwischen Optionsbuchstaben und Parameter darf kein Leerzeichen stehen. Leerzeichen müssen hingegen zur Trennung der Optionen voneinander benutzt werden.

-ennnnnn lädt PEPS erst ab EPROM-Adresse **nnnnnn** (hex, default 000000).

-w<Name> Ausgabe nicht auf PEPS, sondern binär nach Datei **<Name>** oder Auslesen des PEPS und speichern in der Datei **<Name>**, wenn keine Quelldatei angegeben wurde.

-bn bestimmt die Anzahl der angeschlossenen PEPS-Geräte: 1, 2 oder 4 (default 1).

-pn bestimmt die Nummer der Parallel-Schnittstelle: LPT1 (default), LPT2, LPT3 oder LPT4.

-fn Dateiformat (0 = Binär, 1 = INTEL-HEX, 2 = MOTOROLA-S).

Wird die -f Option weggelassen, so erkennt PEPS das Format an der Datei-Extension. .HEX = INTEL, .SHX = MOTOROLA, alles andere: binär.

- cn** Check/Verify-Option: Überprüfung des Datentransfers zum PEPS.
- 0 = Keine Überprüfung.
 - 1 = Nur Hardware-Test durchführen(default).
 - 2 = Alle übertragenen Daten zur Überprüfung zurücklesen.
- on** Debug-Modus: 0 = nicht aktivieren (default), 1 = nach dem Laden aktivieren.
- inn nn..** Patchen: Lädt direkt Bytes (hex) an die mit **-ennnnnn** spezifizierte EPROM-Adresse.

Alle Zahlen (nnnnnn) sind bis zu 6-stellige hexadezimale Eingaben (Ziffern von O-F). Führende Nullen dürfen weggelassen werden. Bei der Option **-i** müssen die dem ersten folgenden Bytes durch jeweils ein Leerzeichen voneinander getrennt sein. Die Reihenfolge der Optionen ist beliebig, lediglich der Dateiname sollte als erstes vor den Optionen stehen.

Ein Beispiel für eine Eingabe sei also:

```
PEPS c:\source\eprom.HEX -wc:\source\eprom.BIN
```

Die INTEL-HEX-Datei C:\source\eprom.HEX wird in eine Binärdatei umgewandelt und nach C:\source\eprom.BIN geschrieben.

Oder:

PEPS eprom.BIN -e0f00 -c2 -p2

Die Datei eprom.bin aus dem aktuellen Inhaltsverzeichnis wird zu PEPS gesendet, und zwar beginnend bei Adresse 0f00 (hex). PEPS ist dabei an den **Druckerport LIT2** angeschlossen (-p2). Jedes Byte wird bei der Übertragung zurückgelesen und geprüft (-c2).

Oder:

PEPS -wD:\read. bin -120000

Liest 128 kB aus dem PEPS aus und speichert sie in der Datei D:\READ.BIN.

Oder:

PEPS -SO 45 50 53 Od Oa -e1000

sendet die Bytes 'P','E','P','S',<CR>,<LF> an das Zielgerät ab Adresse 1000 (hex).

3.1 Anmerkung zu INTEL-HEX Dateien

Endet ein Dateiname mit der Extension .HEX, erkennt PEPS auf INTEL-HEX Datenformat. Mit Hilfe der Segmentadresse, des Startrecords, der Extended-Adress-Records und des Adress-Offsets jedes Datenrecords können diese Daten an die entsprechende Stelle im PEPS übertragen werden. Dabei werden nur die Adressen im PEPS überschrieben, die in den Datenrecords vorkommen, andere Bereiche, auch Lücken zwischen Datenrecords, bleiben unverändert.

Falls die gelesene Adresse ($\text{AdressOffset} + \text{Segment}/16$) größer als 512K ist, wird eine Warnung ausgegeben, da PEPS nur bis 512K Adressbereich simuliert.

Falls ein Record eine falsche Checksumme oder eine falsche Bytezahl enthält bzw. unvollständig ist, wird die Übertragung mit einer entsprechenden Fehlermeldung abgebrochen.

3.2 Anmerkung zu **MOTOROLA-S** Dateien

Endet ein Dateiname mit der Extension **.SHX**, erkennt das PEPS-Ladeprogramm automatisch auf **MOTOROLA-S-Datenformat**. Ähnlich wie im INTEL-HEX Format werden die Ladeadressen automatisch berechnet und Daten an diese Adressen übertragen. Auch dieselben Fehlerbedingungen werden abgetestet und gegebenenfalls angezeigt.

3.3 Anmerkung zu Binärdateien

Wird keines der zwei genannten Formate erkannt, so nimmt der PEPS-Lader an, daß es sich um eine formatlose Binärdatei handelt und überträgt die Daten unverändert ab der mit **-ennnnnn** eingestellten Zieladresse.

Mit der Option **-f0** kann erzwungen werden, jede Datei als Binärdatei aufzufassen. Damit können auch Dateien, die zufällig mit einer Kennung beginnen, aber Binärdatendarstellungen, ordnungsgemäß übertragen werden. Die Optionen **-f1** und **-f2** versuchen eine Datei auch ohne entsprechende Extension als INTEL-HEX bzw. MOTOROLA-S Format zu lesen.

3.4 Ausgabe der Daten in eine Datei

Wird die Option **-w** gewählt, so findet keine Übertragung zum PEPS statt, hingegen werden die Daten in eine Binärdatei auf Diskette umgeleitet. Bei der Ausgabe von INTEL-HEX und MOTOROLA-S Dateien auf Diskette werden Datenlücken zwischen Datenrecords mit NULL-Bytes (00 hex) aufgefüllt. Bei großen Adresssprüngen innerhalb des Files kann es deshalb leicht zur Erzeugung von riesigen Dateien kommen. Vor der Offsetadresse des ersten Datenrecords werden jedoch keine NULL-Bytes aufgefüllt, die Ausgabedatei beginnt immer mit den Daten des ersten Records.

Im Falle eines **Datenrecords** mit niedrigerer Offset-Adresse als der im vorangegangenen **Record** werden die Daten im File angehängt und eine diesbezügliche Warnung ausgegeben.

3.5 Patchen

Mit der Option **-i** können, ohne Angabe einer Datei, eines oder eine Anzahl von Bytes an PEPS gesendet werden. Damit lassen sich einzelne Bytes an beliebigen Adressen verändern, ohne das gesamte Programm erneut zu laden. Die Übertragung geschieht immer auf die mit **-ennnnnn** eingestellten PEPS-Adresse und umfaßt so viele aufeinanderfolgende Bytes, wie die Eingabe zuläßt. Zu beachten sei noch: eine gleichzeitige Angabe einer Lesedatei bewirkt, daß zuerst die Einzelbytes gesendet werden und anschließend die Datei übertragen wird, die dabei erstere möglicherweise überschreibt.

3.6 Datensicherheit

Das PEPS-Ladeprogramm bietet die Möglichkeit, jedes ausgesendete Byte auf Übertragungsfehler hin zu kontrollieren. Mit der **-c2** Option kann dieser Check aktiviert werden, was eine höhere Übertragungszeit zur Folge hat. Andererseits bietet es aber die Sicherheit, daß nicht die Übertragung zum PEPS Schuld am eventuellen Nichtfunktionieren des Zielgerätes ist.

Einen ausreichenden Test der Hardware bietet die **Defaulteinstellung -c1**, bei der nur das erste gesendete Byte rückgelesen wird, alle anderen jedoch nur geschrieben werden.

-c0 schaltet beide Checks aus und überträgt alle Daten ohne irgendwelche Kontrollen, sogar wenn kein PEPS angeschlossen ist.

3.7 Debug-Modus

Dieser Modus hilft bei der Inbetriebnahme eines Ziel-Systems mit ungetesteter Hardware. Er ist insbesondere dann von Wert, wenn zunächst gar nichts funktioniert. Mit **-01** wird der Modus aktiviert,

auf dem Bildschirm erscheinen nach dem Laden die entsprechenden Anweisungen.

Im Debug-Modus läßt sich über die PC-Tastatur der Reset des Zielsystems auslösen, rückzusetzen oder mit wahlbarer Frequenz 'takten'. In dieser Takt-Betriebsart kann mit Hilfe eines **Oszilloskops** das Startverhalten des Zielsystems überprüft werden. Die meisten Hardware-Fehler lassen sich so schnell aufspüren.

Die rote Reset-Klemme des PEPS muß hierfür mit dem **Reset-Eingang** des Zielsystems verbunden werden. Die Klemme führt auf einen **Open-KollektorAusgang**, der bei aktivem Reset gegen GND geschaltet ist. Den Zustand können Sie im Debug-Modus an der LED-Anzeige des PEPS erkennen: Bei leuchtender LED ist der RESET aktiv.

Folgende Tasten sind nach dem Laden im Debug-Modus aktiv:

| | |
|------------------|---|
| ^C,esc,cr | => Rückkehr zum Betriebssystem. |
| R | => Permanent-Reset ein. |
| l | => Getakteter Reset ein. |
| <,> | => Taktfrequenz reduzieren/erhöhen. |
| andere | => Kurzer Reset-Impuls; Takt-Reset aus. |

Bei der Rückkehr zum Betriebssystem bleibt der letzte Zustand der Reset-Leitung erhalten.

3.8 16- und 32-Bit-Zielsysteme

In Zielgeräten mit einem breiteren Datenbus als 8-Bit werden 2 (bei **16-Bit**) oder 4 (bei **32-Bit**) **PEPS**e parallel benötigt, um den Datenbus abzudecken. Mit der Einstellung **-b2** bzw. **-b4** wählen Sie diese Konfiguration aus. Sie bedingt, daß nur jedes 2. (bzw. 4.) Byte reihum an den gleichen PEPS geladen wird, so daß ein **16-Bit** (bzw. **32-Bit**) **EPROM-Datenfile** korrekt zwischen den verschiedenen Simulatoren aufgeteilt wird.

Für diese Betriebsart ist ein **14-poliges** Stufenkabel erforderlich, mit dem die PEPS-Geräte über den Erweiterungsanschluß parallel geschaltet werden. Das Stufenkabel ist ein Flachbandkabel mit vier Pfostenverbindern, bei dem durch Weglassung von Adern Steckplatznummern vergeben werden. Es als Zubehör lieferbar. Der Erweiterungsanschluß hat 14 Pins und ist auf der rechten Seite des PEPS herausgeführt. Stecken Sie das Stufenkabel so auf, daß jeweils gleiche Pole der PEPS-Geräte miteinander verbunden werden (Bus-Prinzip) und Ader 1 des Stufenkabels (markierte Ader) vom PC-Anschluß weg zeigt.

Durch die Verbindung der zwei bzw. vier PEPSs mit dem Stufenkabel vergibt man automatisch Gerätenummern, die vom Ladeprogramm zur Zuordnung der Daten zu den PEPS-Geräten verwendet werden. Der PEPS mit der Nummer Eins befindet sich am ersten Steckplatz des Stufenkabels, das man **am Vorhandensein** aller Bandkabeladern erkennt. Von Steckplatz zu Steckplatz erhöht sich die Gerätenummer jeweils um eins.

Es sind ausschließlich folgende Konfigurationen möglich:

- a) ein PEPS, kein Stufenkabel. S-Bit Modus.
- b) je ein PEPS auf Steckplatz 1 und 2. **16-Bit** Modus.
- c) je ein PEPS auf Steckplatz 1 bis 4. **32-Bit Modus**.

Es darf nur der erste PEPS mit dem PC verbunden sein. An allen anderen PEPS-Geräten müssen evtl. gesteckte D-SUB-Kabel abgezogen werden. Andernfalls **kann** das Ladeprogramm die verschiedenen PEPSs nicht korrekt adressieren.

Falls es beim Laden der PEPS-Geräte zu Adressproblemen kommen sollte, erhalten sie die Meldung

Pepe n nicht ansprechbar.

Falls es trotz korrekter Verbindung der PEPS_e untereinander und zum PC zu Ladefehlern kommt, sollte geprüft werden, ob

- a) alle PEPS_e mit Spannung versorgt werden.
- b) die DIL-Schalter aller PEPS_e korrekt eingestellt sind.
- c) die Pins 18 - 25 der Centronics-Schnittstelle ihres PC's alle mit GND verbunden sind.

4

Steckerbelegung

PEPS ~~wird~~-Stecker zum EPROM-Sockel eine oder mehrere 24- bzw. 28-polige Sockel dazwischen zu stecken; die Pins 1,2,31,32 bzw. 1,2,3,4,29,30,31,32 bleiben frei. (Bei den 32-poligen EPROM-Typen ist es ebenfalls ratsam, einen Sockel unter den Flachbandkabel-Stecker zu stecken, um Beschädigungen an den Stiften zu vermeiden).

Für die Verbindung zur Parallelschnittstelle des Entwicklungssystems (PC) ist das Anschlußkabel mit dem D-SUB-Stecker vorgesehen. Er hat folgende Belegung:

| Pin # | Signal | Bedeutung | Druckerschnittstelle |
|-------|--------|--------------|----------------------|
| 2 | M0 | Modus-Flag | DATA0 |
| 3 | M1 | Modus-Flag | DATA1 |
| 4 | M2 | Modus-Flag | DATA2 |
| 5 | DATA | Datenleitung | DATA3 |

| | | | |
|----|------|------------------------------|-------|
| 6 | CLK1 | Taktleitung 1 | DATA4 |
| 7 | CLK2 | Taktleitung 2 | DATA5 |
| 8 | CLK3 | Taktleitung 3 | DATA6 |
| 9 | CLK4 | Taktleitung 4 | DATA7 |
| 11 | DOUT | Rückleitung gelesener Daten. | BUSY |

18-25 GND Masse-Anschluß

ACHTUNG - das GND-Potential des Zielsystems ist über PEPS mit dem GND des PC verbunden. Wenn das Zielsystem geerdet ist, muß der PC am gleichen Stromnetz betrieben werden. Potentialunterschiede durch beiderseitige Netzerdung können PEPS, Zielsystem und PC zerstören!

5 Die Hardware-Programmierung von PEPS

Dieses Kapitel ist nur **dann** für Sie interessant, wenn Sie Ihr eigenes **PEPS-Ladeprogramm** schreiben wollen. Die Programmierung von PEPS erfolgt über die acht Steuerbits **CLOCK1 - 4, /DATA, MO - M2**, die an acht Ausgangsleitungen des PCs anzuschließen sind, z.B. an eine **PIO** oder eine **Centronics-Schnittstelle**.

Über die Eingänge **MO, M1** und **M2** wird der Betriebsmodus von PEPS eingestellt. Es gibt fünf Modi:

| | | |
|----------------|------------|--|
| M2-1-0: | 000 | Adresse hochzahlen (Mod. 0) |
| | 001 | Datenbyte senden bzw. empfangen (Mod1) |
| | 010 | Byte ins RAM laden (Mod 2) |
| | 011 | EPROM bzw. RAM simulieren (Mod 3) |
| | 100 | Byte vom RAM laden (Mod 4) |

In den Modi 0 bis 2 und 4 liegt der **RESET**-Ausgang auf **GND**, das Zielsystem ist zurückgesetzt, nur im Modus 3 wird der **RESET** inaktiv.

Die **CLOCK**-Leitung löst die durch **MO-2** vorgewählte Aktion aus. **CLOCK** ist **HIGH**-aktiv und muß im Ruhezustand oder beim Modus- Wechsel einen **LOW**-Pegel annehmen.

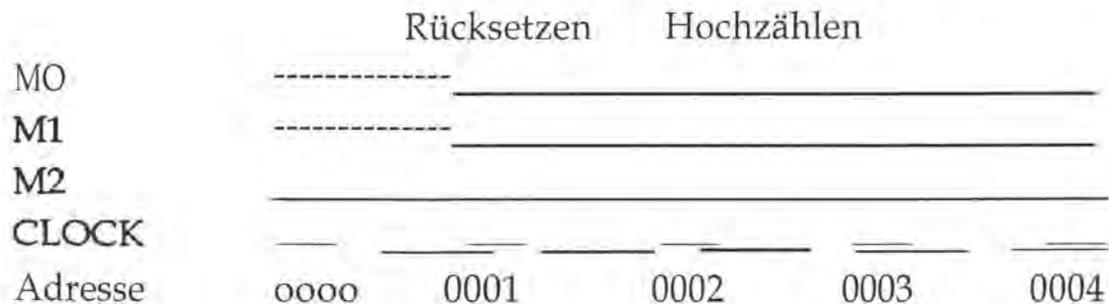
Dabei bedeutet **CLOCK1=>Gerät1** wird angesprochen; **CLOCK2=>Gerät2** usw. (Die Geräte-Nr. wird mit Jumper am PEPS eingestellt)

Im Modus 1 wird über die **/DATA**-Leitung das Datenbyte seriell invertiert eingeschrieben bzw. über die **/DOUT**-Leitung ausgelesen.

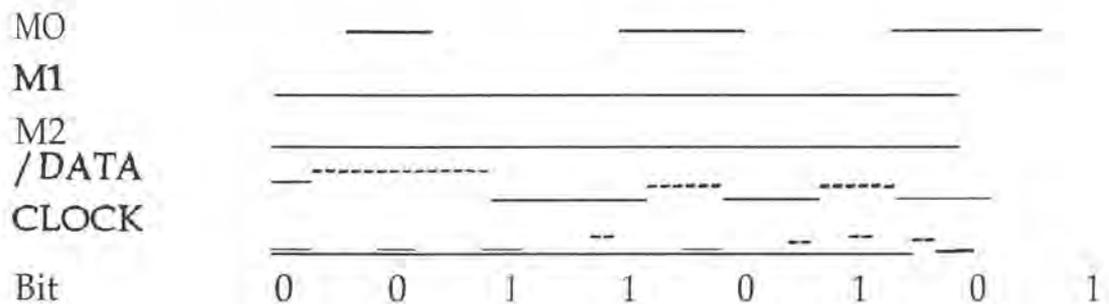
Um ein Datenbyte an eine beliebige PEPS-Adresse zu schreiben, sind folgende Schritte erforderlich:

- 5.1 Adresse einstellen: Im Modus 3 setzt ein HIGH-Pegel am CLOCK-Eingang den internen Adresszähler auf Null. Im Modus 0 wird die eingestellte Adresse bei jeder HIGH-Flanke *von* CLOCK um eins erhöht. Durch Rücksetzen und wiederholtes Hochzahlen läßt sich so jede beliebige Adresse von 0..FFFF (hex) einstellen. Die einmal eingestellte Adresse bleibt auch bei Moduswechsel erhalten.

Das folgende Timing-Diagramm zeigt als Beispiel das Einstellen der Adresse 0004:

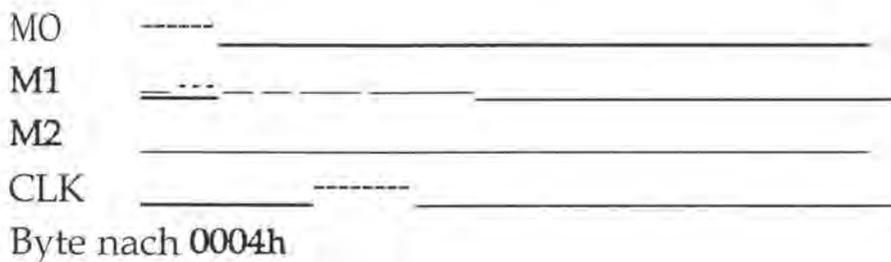


- 5.2 Datenbyte senden: Dazu muß Modus 1 gewählt werden. Nun wird das invertierte Byte seriell über die DATA-Leitung übertragen. Bei jedem LOW-Flanke von CLOCK wird dabei der invertierte Pegel am DATA-Eingang als Bit übernommen. Bit 7 wird zuerst, Bit 0 zuletzt übertragen. Ein einmal eingeschobenes Byte bleibt beim Moduswechsel erhalten. Das dargestellte Impulsdiagramm zeigt das Einschreiben des Bytes 35 (hex) = 00110101 (binär).

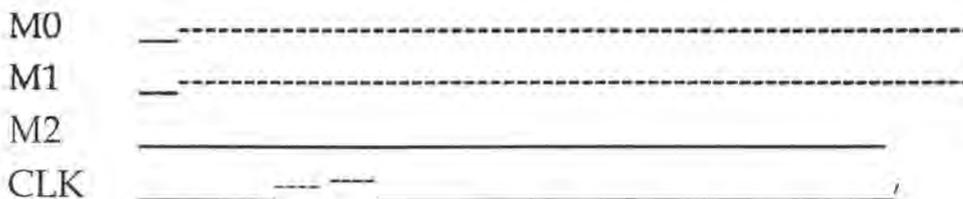


Anmerkung: Die eingeschriebenen Daten erscheinen auf der /DOUT-Leitung um 5-Bit versetzt. Diese Funktion wird zum Prüfen der korrekten Datenübertragung genutzt.

- 5.3 Datenbyte laden: Das in Modus 1 eingeschobene Byte steht zunächst in einem Schieberegister und muß noch ins RAM geladen werden. Das erfolgt in Modus 2. Das Byte wird mit der HIGH-Flanke am CLOCK-Eingang die in Modus 0 eingestellte Adresse geschrieben.



Das Ganze wird nun so lange wiederholt, bis alle Daten übertragen sind. Nach Beendigung des Datentransfers kann der **Simulier-**Modus - Modus 3 - eingestellt werden. Er wird mit einem HIGH-Pegel am CLOCK-Eingang aktiviert. PEPS verhält sich nun wie em EPROM, das mit den entsprechenden Daten "gebrannt" wurde.



Ab jetzt wird simuliert.

Zur Beschleunigung der Übertragung lassen sich einige Schritte abkürzen. Da normalerweise die Daten blockweise übertragen werden, kann man das Einstellen der Adresse auf einfaches Hochzahlen nach jedem übertragenen Byte beschränken. Die Reihenfolge ist also Senden - Laden - Hochzahlen usw. Sollen identische Daten - z.B. Nullen oder FF's - in aufeinanderfolgende Speicherzellen geschrieben werden, genügt einmaliges Senden des Bytes, gefolgt von wiederholtem Laden und Hochzahlen.

5.4 RAM-Simulation

Um auch Daten aus dem Arbeitsspeicher des Zielsystems in den PC zu lesen, besteht auch die Möglichkeit der RAM-Simulation.

Dazu wird nach Einstellung der DIL-Schalter für die entsprechende RAM-Type als erstes der Simulier-Modus - Modus 3 - eingestellt.

Das Zielsystem kann jetzt den PEPS wie einen normalen RAM lesen und beschreiben.

Zum Einlesen der RAM-Daten in den PC wird dann entsprechend 5.1 die zu lesende Adresse eingestellt.

Im Modus 4 -Byte aus RAM laden - wird ein Datenbyte in das Schieberegister übernommen.

M0 _____
 M1 _____
 M2 _____
 CLK _____

Byte übernommen

Das invertierte Byte wird im Modus 1 (entsprechend 5.2) seriell zum PC über die /DOUT-Leitung übertragen. Bei jeder High-Blanke von CLOCK wird dabei der invertierte Pegel am /DOUT-Ausgang übergeben (z.B. 35 H) .