

**NAME**

pep-8 – Beschreibung der pep-8 Architektur

**BESCHREIBUNG**

Der **pep-8** (Practical Education Processor based on pdp-8) ist von der **DEC pdp-8** abgeleitet, und soll der Vermittlung grundlegender Konzepte von Rechnerarchitekturen und Assemblerprogrammierung dienen.

**PROGRAMMIERMODELL**

Der extrem einfach aufgebaute pep-8 Prozessor besteht aus einem Programmzähler (Program Counter, PC) und einem Akkumulator (Accumulator, AC), die zwölf Bit breit sind sowie dem Verbindungsbit (Link Bit, L). Der Speicher ist ebenfalls 12 Bit breit und wird mit 12 Bit Adressen angesprochen. Er umfasst also 4096 12-Bit Worte. Andere Einheiten als ein 12-Bit Wort können nicht adressiert werden. Die Bits eines Wortes werden von null bis elf durchnummeriert, wobei Bit null das signifikanteste Bit ist:

0	1	2	3	4	5	6	7	8	9	10	11
---	---	---	---	---	---	---	---	---	---	----	----

**ADRESSIERUNGSARTEN**

Bei allen Befehlen, die einen Hauptspeicheroperanden haben, wird dessen Adresse in Bit fünf bis elf codiert. Der Speicher ist logisch in Seiten zu 128 Worten aufgeteilt und eine Operandenadresse ist immer relativ zum Beginn der Speicherseite, in der sich der Befehl befindet, oder der Seite null ("Zero Page"). Welche dieser Alternativen gewählt wird entscheidet das "Zero Page" Bit, das im Befehl in Bit vier gespeichert wird. Ist es null, wird relativ zur Seite null adressiert, sonst relativ zur laufenden Seite. Wird das Indirekt-Bit gesetzt, das sich in Bit 3 befindet, so wird der Operand als 12-Bit Zeiger auf ein Speicherwort betrachtet, auf das sich der Befehl dann bezieht. Das ergibt die folgenden Adressierungsarten:

Name	I	ZP	Bits	Erreichbare Worte
Seitenrelativ	0	1	7 Bit	In der aktuellen Seite
Zero Page	0	0	7 Bit	In der nullten Seite
Indirekt	1	1	12 Bit	Im gesamten Speicher, Zeiger in der aktuellen Seite
Indirekt Zero Page	1	0	12 Bit	Im gesamten Speicher, Zeiger in der nullten Seite

Die effektive Adresse für jeden Befehl mit einem Hauptspeicheroperanden berechnet sich aus der Adresse im Befehl (7 Bit, A0-7) und der effektiven Seite (effective page, P0-4). Die effektive Seite ist Null wenn das Zero-Page-Bit null ist, und entspricht sonst den fünf signifikantesten Bits des Programmzählers.

0	1	2	3	4	5	6	7	8	9	10	11
P0	P1	P2	P3	P4	A0	A1	A2	A3	A4	A5	A6

Ist das Indirekt-Bit gesetzt wird dieser Wert noch dereferenziert, d.h. der Inhalt des durch ihn adressierten Speicherwortes bildet die effektive Adresse. Wird nicht indirekt adressiert, bildet der Wert aus effektiver Seite und Adresse im Befehlsword direkt die effektive Adresse.

**BEFEHLE**

Der pep-8 Prozessor verfügt über sieben Befehle die auf dem Hauptspeicher operieren. Bei diesen wird der Operationscode in den Bits null bis drei abgelegt. Zu jedem Befehl sind die Mnemonische Abkürzung, der englische Begriff aus den sich diese ableitet, sowie der Operationscode in Binärdarstellung aufgeführt.

**RCL** (Recall, 000)

Läd das im Hauptspeicher adressierte Wort in den Akkumulator.

**STO** (Store, 001)

Speichert den Akkumulator im adressierten Speicherwort.

**AND** (And accumulator, 010)

Der Akkumulator wird mit dem adressierten Wort bitweise und-verknüpft. Das Ergebnis wird in den Akkumulator zurückgeschrieben.

**ORA** (Or accumulator, 011)

Der Akkumulator wird mit dem adressierten Wort bitweise oder-verknüpft. Das Ergebnis wird in den Akkumulator zurückgeschrieben.

**TAD** (Two's Complement Add, 100)

Der Operand wird zu dem 13-Bit Wert der durch das Verbindungsbit und Akkumulator gebildet wird (L, AC) addiert. Negative Zahlen werden im Zweierkomplement repräsentiert.

**JMP** (Jump, 101)

Das adressierte Hauptspeicherwort wird in den Programmzähler geladen. Es wird also zur angegebenen Adresse gesprungen.

**JMS** (Jump to Subroutine, 110)

Der Programmzähler wird im adressierten Hauptspeicherwort gespeichert. Dann wird die Adresse des darauf folgenden Speicherwortes in den Programmzähler geladen. Es wird also tatsächlich zu der Adresse gesprungen die auf die angegebene Adresse folgt. Ein Rücksprung erfolgt dann über einen indirekten Sprung mit der Startadresse der Routine als Argument.

Die verbleibende Bitkombination *111* (OPR, Operate) identifiziert einen der restlichen Befehle des pep-8, die keinen Hauptspeicheroperanden benötigen und deshalb die Bits drei bis elf zur Bestimmung des Befehls heranziehen können. Zu jedem Befehl wird hier der vollständige 12-Bit Befehlscode als Oktalzahl angegeben.

*Befehle die den internen Zustand ändern*

Diese Befehle können in einem Befehlswort kombiniert werden. Bei diesen Befehlen ist Bit 4 auf null.

**NOP** (No Operation, 7000)

Nichtstun. Kann mit keinem anderen Befehl kombiniert werden.

**CLL** (Clear Link Bit, 7100)

Verbindungsbit auf Null setzen.

**STL** (Set Link Bit, 7120)

Verbindungsbit auf Eins setzen.

**CML** (Complement Link, 7020)

Verbindungsbit invertieren.

**CLA** (Clear accumulator, 7200)

Akkumulator auf Null setzen.

**STA** (Set accumulator, 7240)

Alle Bits des Akkumulators auf Eins setzen.

**CMA** (Complement accumulator, 7040)

Einerkomplement des Akkumulators bilden und dort ablegen.

**IAC** (Increment accumulator, 7001)

Akkumulator um eins erhöhen.

**RAL** (Rotate left Link and accumulator, 7004)

Den 13-Bit Wert Verbindungsbit-Akkumulator (L+AC) nach links rotieren.

**RAR** (Rotate right Link and accumulator, 7010)

Den 13-Bit Wert Verbindungsbit-Akkumulator (L+AC) nach rechts rotieren.

Die sicherlich wichtigste Befehlskombination dient der Subtraktion. Man lädt den Subtrahenden in den Akkumulator und wendet den kombinierten Befehl **CLL CMA IAC** (7141) an, der das Link-Bit löscht und das Zweierkomplement bildet. Anschließend wird mit **TAD** der Minuend addiert und die Differenz steht im Akkumulator.

*Befehle die den Programmfluß verändern*

Bei diesen Befehlen ist Bit 4 gesetzt und Bit 11 auf null. Diese Befehle können in einem Befehlswort kombiniert werden, solange die Befehle zum gleichen Kombinationstyp gehören. Die Sprungbedingungen werden oder-verknüpft, wenn Bit 8 des Befehlswortes null ist (*Oder-Typ*). Ist Bit 8 eins, werden sie und-verknüpft (*Und-Typ*).

**HLT** (Halt the program, 7402)

Das Programm anhalten.

**SNL** (Skip on non-zero Link, 7420, Oder-Typ)

Wenn das Verbindungsbit nicht null ist, den Programmzähler erhöhen.

**SZA** (Skip on zero accumulator, 7440, Oder-Typ)

Wenn der Akkumulator null ist, den Programmzähler erhöhen.

**SMA** (Skip on minus accumulator, 7500, Oder-Typ)

Bei negativem Akkumulator den Programmzähler erhöhen.

**SKP** (Skip always, 7410, Und-Typ)

Den Programmzähler erhöhen (Und-Typ).

**SZL** (Skip on zero Link, 7430, Und-Typ)

Wenn das Verbindungsbit null ist, den Programmzähler erhöhen.

**SNA** (Skip on non-zero accumulator, 7450, Und-Typ)

Wenn der Akkumulator nicht null ist, den Programmzähler erhöhen.

**SPA** (Skip on plus accumulator, 7510, Und-Typ)

Bei positivem Akkumulator den Programmzähler erhöhen.

*Befehle zum Ansprechen von Ein-/Ausgabegeräten*

Mit diesen Befehlen können Ein-/Ausgabegeräte gesteuert werden. Der pep-8 Prozessor unterstützt bis zu 16 Geräte. Jedes Gerät besitzt eine oder mehrere Untereinheiten mit je einem Ein- und/oder Ausgabekanal. Bei diesen Befehlen sind Bit 4 und Bit 11 gesetzt. Das jeweils angesprochene Gerät wird im Befehlswort in den Bits 7-10 kodiert, entsprechend muss die Gerätenummer eine Stelle nach rechts geschoben werden (verdoppeln) bevor sie mit dem Befehlswort verodert wird. Beispiel: Lesen von Gerät 3: **DGA (3\*2)**. Diese Befehle können nicht kombiniert werden.

**SRI** (Skip on ready for input, 7401)

Den Programmzähler erhöhen wenn der Eingabekanal bereit ist.

**SRO** (Skip on ready for output, 7441)

Den Programmzähler erhöhen wenn der Ausgabekanal bereit ist.

**DGA** (Device get word to accumulator, 7501)

Den Akkumulator mit dem aktuellen Wert des Eingabekanals laden.

**DPA** (Device put word from accumulator, 7541)

Den Wert des Akkumulators in den Ausgabekanal schreiben.

**DUS** (Device unit select, 7601)

Untereinheit gemäß dem Wert im Akkumulator auswählen.

**DGS** (Device get status word, 7641)

Statuswort des Geräts bzw. der Untereinheit im Akkumulator speichern.

**DSM** (Device sense mask, 7701)

Den Wert des Akkumulators als Meldungsauswahlmaske an das Gerät übergeben.

**RSD** (Reset device, 7741)

Das Gerät zurücksetzen und Untereinheit null auswählen.

## GERÄTE

*Teletype, Gerät 0 (Null)*

In Verbindung mit dem Terminalsimulator **teletype**(1) kann **pepsi**(1) Zeichen ein- und ausgeben, wenn das Gerät beim Start freigeschaltet wird. Das Teletype (TTY) unterstützt die Befehle **SRI**, **SRO**, **DGA** und **DPA**.

*Papertape Reader, Gerät 1*

Dem Gerät wird beim Freischalten eine Datei zugewiesen aus der Einzelzeichen gelesen werden können. Am Dateiende geht das Gerät in den nicht-bereit Zustand. Der Papertape Reader unterstützt die Befehle **SRI** und **DGA**.

## SIEHE AUCH

**pot**(1), **pot**(5), **pepsi**(1)