

## **Monitorkommandos AC1 - 1985**

**Das Betriebssystem des Ur-AC1 und damit dessen Bedienung ist grundsätzlich erhalten geblieben. Im Folgenden sind die Kommandos zusammengefasst:**

### **Arithmetik**

#### **A aaaa bbbb c (Arithmetik)**

Es werden Summe, Differenz, wenn möglich die relative Distanz für einen Sprungbefehl aus den ersten beiden Argumenten sowie der dezimale Wert des ersten Arguments, berechnet. c gibt die Länge des Sprungbefehls an.

### **Breakpoint**

#### **B aaaa**

Dieser Befehl ist nützlich für das Testen von bzw. die Fehlersuche in Programmen. Er setzt einen Softwarehaltepunkt auf die Adresse aaaa.

aaaa muß dabei immer auf das erste Byte eines Befehls zeigen. Ist nach dem Starten des Programms diese Adresse erreicht, erfolgt die Ausschrift „BREAK AT aaaa #“ und die Kontrolle geht wieder an den Monitor zurück. Zuvor werden alle Registerinhalte der CPU in die RSA (Register Save Area) gerettet, sodass sie sich z.B. mittels des R-Befehls anzeigen lassen. Das zu testende Programm wird durch den Break-Befehl nicht zerstört. Es lässt sich mit dem Go-Befehl fortsetzen.

### **Compare**

#### **C aaaa bbbb cccc**

Dieser Befehl vergleicht die beiden Speicherinhalte ab aaaa und bbbb für die Länge von cccc Bytes miteinander. Wird eine Ungleichheit gefunden, erscheinen die beiden nicht übereinstimmenden Bytes mit den jeweils zugehörigen Adressen auf dem Schirm. Mit der CR-Taste kann man die Suche fortsetzen. Jede andere Taste führt zum Abbruch.

### **Memory-Dump**

#### **D aaaa bbbb**

Der Speicherinhalt von der Adresse aaaa bis zur Adresse bbbb wird als Hexdump auf dem Bildschirm ausgegeben. Zu Beginn jeder Zeile erfolgt die Ausgabe der jeweiligen Anfangsadresse, es folgen 16 Datenbytes. FFFFh als Endadresse für zu einem Overrun und einem endlosen Durchscrollen (Bug von 1983).

### **Execute**

#### **E aaaa**

Das Maschinenprogramm ab Adresse aaaa läuft unter Breakpointkontrolle. Das heißt, die Register der CPU werden entsprechend der aktuellen Werte der RSA geladen, der Softwarebreakpoint wird aktiviert und zur Adresse aaaa gesprungen.

### **Find string**

#### **F aa bb cc dd ... nn**

Hier ist ab Adresse 00aaH die Datenfolge aa bb cc dd ... nn zu suchen. Die zu suchende Datenfolge oder Zeichenkette kann dabei maximal so lang sein, daß das gesamte Kommando gerade noch auf den Bildschirm passt. Wird die angegebene Zeichenkette vollständig gefunden, springt der Monitor an den M-Befehl, wobei die Adresse auf das erste Byte der Datenfolge zeigt. Soll die Suche nach dem M-Befehl weitergehen, ist die Folge F: aa bb cc dd ... nn einzugeben. Wird die Datenfolge im gesamten Speicher nicht gefunden, erfolgt die Ausschrift „NOT FOUND #“.

### **Go on**

#### **G**

Dieses Kommando funktioniert analog dem E-Befehl. Die Programmausführung wird hierbei ab der Adresse im PC fortgesetzt. Falls ein Breakpoint zuvor eingegeben wurde, wird dieser aktiviert.

### **Initialize**

#### **/**

Hiermit werden alle Anwenderregister gelöscht. Alle Speicherzellen der RSA werden auf Null gesetzt. Der Stackpointer wird mit 2000h initialisiert.

## **Jump to**

**J aaaa**

Dieser Befehl führt den Ansprung eines Anwenderprogramms ab Adresse aaaa ohne Aktivierung eines Breakpoint aus

## **Landen von Kassette**

**L aaaa ±**

Der Computer versucht, ein Programm oder eine Datei, vom Bandgerätinterface in den Speicher zu laden. Das optionale Argument aaaa± bewirkt einen Speicheroffset beim Einlesen .

Die Startadresse des Programms wird in ARG1 geladen, sodass ein Programm anschließend mit J: gestartet werden kann.

## **Modify Memory**

**M aaaa**

Hiermit kann der RAM-Speicher ab der Adresse aaaa byteweise angezeigt und neu geschrieben werden. Nach jedem Betätigen der CR-Taste erscheinen die aktuelle Adresse und das zugehörige Byte. Nach dem Prompt lässt sich ein neues Byte oder eine Bytefolge (durch Leerzeichen getrennt) eingeben.

Fortlaufendes Betätigen der CR-Taste führt zu einer byteweisen Speicheranzeige. Nach einer Dateneingabe wird der Wert übernommen und geprüft.

Abgeschlossen wird das Kommando durch Eingabe eines Punktes.

## **Pattern**

**P aaaa bbbb cc**

Dieser Befehl füllt den Speicher von der Adresse aaaa bis zur Adresse bbbb mit dem Bitmuster cc.

## **Register anzeigen/ändern**

**R XX**

Dieses Kommando ermöglicht die Anzeige und das Verändern der Inhalte aller CPU-Register. XX steht für das jeweilige Registerpaar.

Nach dem Drücken der CR-Taste erscheint der Wert des jeweiligen Doppelregisters, gefolgt vom # Mit der nachfolgenden Eingabe eines Wertes lässt sich das Registerpaar dann neu setzen.

Folgt dem R ein Doppelpunkt, so werden der gesamte Registersatz, der Breakpoint, die Breakpointsequenz (3 Bytes ab Breakpointadresse), sowie die gesetzten Flaggs angezeigt.

## **Speichern auf Kassette**

**S aaaa bbbb cccc name**

Ein Speicherbereich ab der Adresse aaaa bis zur Adresse bbbb wird mit der Startadresse cccc unter dem Namen name (Länge max 6 Zeichen) auf Magnetband gespeichert.

## **Transfer**

**T aaaa bbbb cccc**

Der Speicherbereich mir cccc Bytes wird ab Adresse aaaa nach Adresse bbbb verschoben. Überlappungen sind zulässig.

## **Vergleichen mit Kassette**

**V**

Ein auf Magnetband gespeichertes File kann mit dem dem Speicher verglichen werden

## **Aufruf Minibasic**

**Z**

Dieser Befehl ist in der Monitorvariante 1 obsolet, da der Speicher ab 0800h mit der Monitorerweiterung belegt ist. In der Monitorvariante 2 ist der Befehl aktiv.