EUMEL

Extendable multi User Microprocessor ELAN-system

Anwendersoftwareklasse 2

Modellcomputer
MOCO

Hochschul-
Rechen-
Zentrum

Universität Bielefeld

GMD

INHALTSVERZEICHNIS

1. Allgemeines

1.1 Zweck, Vorkenntnisse, mögliche Lernziele

MOCO ist ein Modellsystem zur Veranschaulichung der Grundprinzipien der Computertechnik.

Für das Arbeiten mit dem Modellcomputer sind als Vorkenntnisse lediglich Kenntnisse der 4 Grundrechenarten im dezimalen Zahlensystem erforderlich.

Unter Verwendung des Modellcomputers MOCO können folgende prinzipiellen Methoden gelernt werden :

1. Wie ein Computer durch Kommandos gesteuert und bedient wird.
2. Das Daten und Programme im gleichen Speicher gehalten werden (v. Neumann Prinzip).
3. Wie ein Computer zwischen Daten und Programmen unterscheidet (Bedeutung des Befehlszeigers).
4. Wie der Computer ein Programm ausführt (Prinzip des Ausführungszyklus).
5. Wie Berechnungen mit Hilfe des Akkumulators durchgeführt werden (Einadressbefehle).
6. Wie der Computer auf Grund vorgegebener Bedingungen entscheidet (Vergleichsbefehl).
7. Wie der Computer Befehlsfolgen wiederholt (Sprungbefehl und Schleifen).
8. Wie der Computer ein bestimmtes Verarbeitungsschema (d.h. ein Programm) wiederholt und auf verschiedene Eingabedaten anwendet (EVA - Grundprinzip der DV).
9. Wie im Verlauf der Ausführung eines Programms ein Unterprogramm ausgeführt werden kann.
10. Wie Datentabellen elementweise verarbeitet werden können (Adressrechnung und Indizierung).

1.2 Speicheraufbau, Sichtfensterprinzip

MOCO ist ein Modellcomputer mit einem Speicher von 100 Zellen (auch Register genannt). Der Speicher wird in Form einer Tabelle auf den Bildschirm geschrieben und kann durch Ausfüllen programmiert werden (Sichtfensterprinzip). Er enthält Daten und Programme. Die Zellen haben die Adressen 00, 01, 02, ..., 99. Der Speicherinhalt kann auch in eine Datei abgelegt (gerettet) und von dort wieder geladen werden. Jede Speicherzelle kann eine 4 - stellige Dezimalzahl aufnehmen und diese wird als zu verarbeitende Zahl oder als steuernder Befehl interpretiert.

Für den Lernenden ist das Sichtfensterprinzip, das dem Konzept dieses Modellcomputers zu Grunde liegt, wesentlich :

> Alle Informationen, über die der Modellcomputer verfügt, kann auch der Lernende im Sichtfenster sehen und sogar verändern.

Wegen dieses Prinzips sind auch die internen Arbeitsregister des Modellcomputers auf den ersten 3 Zellen des 100 - Zellen - Speichers untergebracht.

Die Speicherzelle 00 enthält den Akkumulator, der Verarbeitungsergebnisse aufnimmt und als Operand in vielen Verarbeitungen verwendet wird.

Die Zelle 01 enthält den Befehlszeiger, d.h. die Adresse der Zelle, deren Inhalt bei dem nächsten Schritt des Modellcomputers als Befehl interpretiert wird.

Die Zelle 02 nimmt bei einem Sprung aus einem Programm an den Anfang eines zweiten Programms die Adresse des nächsten Befehls des 1. Programms auf, um eine spätere Fortsetzung zu ermöglichen.

Aufbau des Sichtfensters/Bildschirmaufbau :

```
      M O C O - MODELLCOMPUTER
      AK   BZ   02   03   04   05   06   07   08   09
  0 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
 10 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
 20 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
 30 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
 40 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
 50 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
 60 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
 70 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
 80 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
 90 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000


AUSGABE :
EINGABE :
DIALOG  :
KOMMANDO:

    MENU: ,  , a, r, l, e, u, v, z, ?, s
```

1.3 Aufträge

Bei der Benutzung des Modellcomputers verwendet man Kommandos, um sofort bestimmte Aktionen zu veranlassen, wie z.B. das Laden eines Programms aus einer Datei, oder das Starten der Ausführung eines Programms.

Der Kommandocode besteht aus genau einem Buchstaben. Manche Kommandos erfordern noch weitere Angaben. Z.B. erfordert das RETTE - Kommando den Namen, unter dem das Programm abgelegt werden soll.

Befehle sind die elementaren Bestandteile eines Programms. Jeder Befehl ist als 4 - stellige Dezimalzahl in einer Zelle des 100 - zelligen Speichers dargestellt. Bei der Ausführung wird die

1. Ziffer als Befehlscode, die
2. Ziffer als Längen - oder Registercode und die
3. - 4. Ziffer als Adresscode interpretiert.

Es können nur Befehle ausgeführt werden, die im Speicher abgelegt sind. Die Ausführung einzelner Befehle oder auch einer ganzen Befehlsfolge wird durch Kommandos gestartet.

1.4.Aufruf von MOCO

Der Modellcomputer wird im Dialog mit dem EUMEL - Monitor nach der Aufforderung

gib kommando :

durch die Eingabe: moco
eingeschaltet. Vergessen Sie bitte nicht nach dieser Eingabe die RETURN Taste zu betätigen.

2. KOMMANDOS

2.1 Kommandoübersicht

Kommandos werden in der KOMMANDO – Zeile des Bildschirms gegeben.

Code	Bedeutung
'' –	(Taste: Pfeil nach oben) Schreiben in den Speicher
' ' –	(Taste: Zwischenraum) Ausführen des nächsten Befehls
'a' –	Ausführen des ganzen Programms im Speicher
'i' –	Initialisieren des Speichers
'r' –	Retten des Speicherinhalts in eine Datei
'l' –	Laden des Speichers aus einer Datei
'e' –	Editieren einer Datei
'u' –	Übersetzen eines symbolischen Programms
'v' –	Vergessen (Löschen) einer Datei
'z' –	Zeigen des Speichers
'?' –	Auskunft über mögliche Kommandos
's' –	Stop, Abschalten des Modellcomputers

2.2 Kommando – Beschreibungen

2.2.1 '' – Kommando (Schreiben)

Das Schreiben in den Speicher erfolgt unmittelbar dadurch, daß man an den entsprechenden Positionen den Bildschirm beschreibt. Nach Drücken der Positionierungstaste :Pfeil nach oben: in der KOMMANDO – Zeile kann der Speicher überschrieben werden. Dies kann beliebig oft und im Wechsel mit dem Ausführungskommando geschehen. Auf diese Weise ist eine Programmentwicklung und sofortige Erprobung im Bildschirmdialog mit dem Modellcomputer möglich.

Es können nur Ziffern sowie an erster Stelle jeder Speicherzelle ein Minuszeichen eingegeben werden. Das Schreiben wird beendet durch die ESC – Taste bzw. durch Positionierung in die KOMMANDO – Zeile.

2.2.2 ' ' – Kommando (Einzelschritt – Ausführung)

Wenn nur ein Zwischenraum eingegeben wird, also bei

:BLANK: (in der KOMMANDO – Zeile)

wird jeweils der Befehl ausgeführt, auf den der Befehlszeiger zeigt, d.h. der Befehl, dessen Adresse in der Zelle 01 steht.

Dies Kommando wird zur schrittweisen Verfolgung einer Programmausführung empfohlen.

2.2.3 'a' – Kommando (Ausführen)

Nach Eingabe von

a (in der KOMMANDO – Zelle)

wird das im Speicher des MOCO stehende Programm ausgeführt.

Bei der Ausführung eines Programms wird folgender Ausführungszyklus vom Modellcomputer durchgeführt:

1. Befehlszeiger lesen (Zelle BZ)
2. Befehl lesen
3. Befehlszeiger um 1 erhöhen

4. Befehl interpretieren und ausführen

Wie bei einem Viertaktmotor werden diese Schritte ständig wiederholt bis zu
einem programmierten,
einem fehlerbedingten oder
einem erzwungenen STOP.

2.2.4 'i' – Kommando (Initialisieren)

Das Kommando i bewirkt, daß alle Speicherzellen auf 0000 gesetzt werden. Ob wirklich ein Initialisieren (Löschen) des Speichers gewünscht wird, wird in der DIALOG – Zeile abgefragt. Die Inhalte der EINGABE –, AUSGABE –, KOMMANDO – und der DIALOG – Zeile werden bei der Initialisierung gelöscht.

Möglich:	i	(in der KOMMANDO – Zeile)
	j oder n	(in der DIALOG – Zeile)

2.2.5 'r' – Kommando (Retten)

Fertige Programme können in einer Datei aufbewahrt werden. Der Name, unter dem das Programm abgelegt werden soll, wird im Dialog erfragt. Bei der Eingabe von ? erscheint eine Liste aller verfügbaren Dateien. Soll das Programm wieder geladen werden, so muß beim l – Kommando der gleiche Dateiname angegeben werden.

Beispiel:

	r	(in der KOMMANDO – Zeile)
	prim :RETURN:	(in der DIALOG – Zeile)

2.2.6 'l' – Kommando (Laden)

In der DIALOG – Zeile wird der Name des zu ladenden Programms erfragt. Das Programm wird dann aus der entsprechenden Datei geladen und auf dem Bildschirm gezeigt.

Beispiel:

l	(in der KOMMANDO – Zeile)
wurzel :RETURN:	(in der DIALOG – Zeile)

Bei Eingabe von ? wird eine Liste aller verfügbaren Dateien gezeigt.

Bei fehlerhaftem oder unbekanntem Programmnamen wird ein Fehlerkommentar ausgegeben. Nach Drücken der ESC – Taste kann dann ein neues Kommando eingegeben werden.

2.2.7 'e' – Kommando (Editieren)

In der DIALOG – Zeile wird der Name der zu editierenden Datei erfragt. Die Datei wird dann auf dem Bildschirm gezeigt und kann editiert werden. Bei Eingabe von ? wird eine Liste aller verfügbaren Dateien gezeigt.

Beispiel:

e	(in der KOMMANDO – Zeile)
wurzel :RETURN:	(in der DIALOG – Zeile)

Bei fehlerhaftem oder unbekanntem Programmnamen wird ein Fehlerkommentar ausgegeben. Nach Drücken der ESC – Taste kann dann ein neues Kommando eingegeben werden.

2.2.8 'u' – Kommando (Übersetzen)

Nach Eingabe des Kommandos

u (in der KOMMANDO – Zeile)

wird das in einer Datei in symbolischer Form vorliegende Programm übersetzt, im Zifferncode in den Speicher geladen und am Bildschirm gezeigt. Der Name der Datei wird in der DIALOG – Zeile erfragt. Bei Eingabe von ? erscheint eine Liste aller verfügbaren Dateien.

Symbolische Form eines Befehls:

LABEL BEFEHL OPERANDENADRESSE KOMMENTAR

z.B. start ein 55 Erstes Einlesen

Zwischen den einzelnen Befehlsteilen muß mindestens ein Blank stehen. Jeder Befehl muß in einer neuen Zeile beginnen.

Für die Befehle (Ziffern 0 – 9) sind die folgenden mnemotechnischen Abkürzungen vorgesehen:

Zifferncode	mnemotechnischer Code	Funktion
0xxx	abl	Ablegen
1xxx	add	Addieren
2xxx	sub	Subtrahieren
3xxx	mul	Multiplizieren
4xxx	div	Dividieren
5xxx	ein	Eingeben
6xxx	aus	Ausgeben
7xxx	vgl	Vergleichen
8xxx	spr	Springen
9xxx	hol	Holen

0000	stp	Stop, Programm beenden
1000	dpp	AK – Inhalt verdoppeln
2000	nll	AK – Inhalt auf 0 setzen
3000	qua	AK – Inhalt quadrieren
4000	ens	AK – Inhalt auf 1 setzen
9000	nop	No operation, Leerbefehl
7000	skp	Skip, Überspringe den nächsten Befehl

Die Operandenadresse kann als Zahl oder als symbolische Adresse angegeben werden, der eventuell durch Komma getrennt die Nummer des verwendeten Indexregisters oder die Länge bei Ein/Ausgabebefehlen folgt, also z. B.:

55 oder Zahl oder 62,9 oder Start,4

Darüberhinaus wird der Befehl dez verwendet, um den Inhalt einer Zelle als Dezimalzahl zu definieren. Der Befehl org bewirkt, daß die nachfolgenden Befehle ab der angegebenen Zelle abgelegt werden. Mit end wird das Ende des symbolischen Programms angezeigt. Ausserdem können Kommentarzeilen (mit einem * an erster Stelle der Zeile) zur Dokumentation in das Programm eingefügt werden.

Beispiel für ein symbolisches Programm

```
* Addition von 5 Zahlen, Berechnung des Mittelwertes
*
       org 20              Programmbeginn in Zelle 20
Start  ein Zahl,5          5 Zahlen einlesen
       hol Zahl            Zahlen addieren
       add Zahl+1
       add Zahl+2
       add Zahl+3
       add Zahl+4
       abl Summe           Summe merken
       div fuenf           Mittelwert berechnen
       aus Zahl,5          Zahlen ausgeben
       aus Summe           Summe ausgeben
       aus 0,1             Mittelwert ausgeben
       stp                 Programm beenden
* Datenteil
Zahl   dez 0               Bereich für Zahlen
       dez 0
       dez 0
       dez 0
       dez 0
Summe  dez 0               Bereich für Summe
fuenf  dez 5               Konstante für Mittelwert
       end
```

2.2.9 'v' – Kommando (Vergessen)

Nicht mehr benötigte Dateien können vom MOCO aus gelöscht (vergessen) werden. Der Name der Datei wird in der DIALOG – Zeile erfragt. Bei Eingabe von ? erscheint eine Liste aller verfügbaren Dateien.

Beispiel :

v		(in der KOMMANDO – Zeile)
dat1	:RETURN:	(in der DIALOG – Zeile)

2.2.10 'z' – Kommando (Zeigen)

Mit diesem Kommando kann der Benutzer sich den Speicher erneut zeigen lassen.

2.2.11 '?' – Kommando (Auskunft)

Bei Eingabe eines ? erscheint eine Auskunft über mögliche Kommandos am Bildschirm.

durch Drücken der Tasten :ESC: q wird die Auskunft beendet und der Modell-computer arbeitet weiter.

2.2.12 's' – Kommando (Stop)

Das Kommando

s (in der KOMMANDO – Zeile)

schaltet den Modellcomputer ab.

3. BEFEHLE

3.1 Befehlscodes

Code	Bedeutung	Symbolische Schreibweise
0	Ablegen	abl
1	Addieren	add
2	Subtrahieren	sub
3	Multiplizieren	mul
4	Dividieren	div
5	Eingeben	ein
6	Ausgeben	aus
7	Vergleichen	vgl
8	Springen	spr
9	Holen	hol

3.2 Spezialbefehle (wichtige Spezialfälle)

Code	Bedeutung
0000	Stop (Symbolische Schreibweise: stp)
1000	Verdoppeln (dpp)
2000	Null setzen (nll)
3000	Quadrat (qua)
4000	Eins setzen, auch wenn AK = 0 (ens)
5100	Einlesen einer Zahl in den AK
6100	Ausgeben des AK-Inhalts
7000	Überspringen des nächsten Befehls (skp)
8000	Retten des BZ ohne zu springen
8200	Rücksprung
9000	Keine Operation (nop)

3.3 Befehls - Beschreibungen

Jeder Befehl ist als 4 - stellige Dezimalzahl in einer Zelle des 100 - zelligen Speichers dargestellt. Bei der Ausführung wird

die 1. Ziffer als Operationscode,
die 2. Ziffer als Längen - oder Registercode,
die 3. - 4. Ziffer als Adresscode interpretiert.

Der Längencode wird nur bei Eingabe - und Ausgabebefehlen verwendet.

Der Registercode wird bei allen übrigen Befehlen verwendet, sollte jedoch vom Anfänger zunächst noch nicht benutzt werden. Seine Interpretation wird im Kapitel Adressrechnung für alle Befehle gemeinsam behandelt.
In den folgenden Beschreibungen wird Registercode = 0 angenommen.

3.3.1 Befehlscode 0 : Ablegen (abl)

Der Inhalt des Akkumulators (AK = Abkürzung für Akkumulator) wird in die Zelle abgelegt, deren Adresse im Befehl, nämlich im Adresscode angegeben ist.
Beispiel : 0035 - Speichere den AK - Inhalt
in die Zelle mit der Adresse 35

Sonderfall : 0000 - STOP, denn die Zelle 00 wird als AK benutzt.

3.3.2 Befehlscode 1,2,3,4: Rechnen

Bei jeder Berechnung wird der AK - Inhalt und der Inhalt der im Befehl angegebenen Zelle verwendet. Das Ergebnis der Berechnung ersetzt den bisherigen AK - Inhalt.

Es gibt Rechenbefehle für die vier Grundrechenarten Addieren, Subtrahieren, Multiplizieren und Dividieren. Alle Berechnungen werden nur mit ganzen Dezimalzahlen durchgeführt, die bis zu vierstellig sein können.

Zahlen mit weniger als vier Stellen können in der 1.Stelle ein negatives Vorzeichen haben. Ist das Ergebnis einer Berechnung eine Zahl mit mehr als 4 Stellen, werden die letzten vier Stellen abgespeichert.

Beispiele :

1032 – Addiere zum AK – Inhalt den Inhalt der Zelle 32
2017 – Ziehe vom AK – Inhalt den Inhalt der Zelle 17 ab
3086 – Nimm den AK – Inhalt mal dem Inhalt der Zelle 86
4055 – Teile den AK – Inhalt durch den Inhalt der Zelle 55

Spezialfälle:

1000 – Verdopple den AK – Inhalt (addiere zum AK den Inhalt der Zelle 00).
2000 – Lösche den AK (ziehe den AK – Inhalt von sich selber ab).
3000 – Berechne das Quadrat des AK – Inhalts.
4000 – Setze den AK – Inhalt gleich Eins,

3.3.3 Befehlscode 5 : Eingeben

Vom Dialogbenutzer werden in der EINGABE – Zeile die im Längencode des Befehls angegebene Anzahl von Eingabedaten eingelesen und in die im Befehl (nämlich im Adresscode) angegebene und in nachfolgende Zellen gebracht.

Falls nicht genügend Daten eingegeben wurden, werden weitere nachgefordert.

Der Dialogbenutzer darf nur Zahlen eintippen, sowie jeweils einen Zwischenraum, der die Zahlen voneinander trennt. Bei Eingabe von unzulässigen Zeichen muß die Eingabe wiederholt werden.

Die Beendigung der Eingabe erfolgt normalerweise durch einen Zwischenraum, sie kann aber durch die RETURN - Taste abgebrochen werden (= Programm-abbruch).

Beispiele: 5123 liest eine Zahl und bringt sie in die Zelle mit der Adresse 23.
5468 liest vier Zahlen und bringt sie in die Zellen 68,69,70,71.
Speziell: 5100 liest eine Zahl in den AK.
Sonderfall: 50xx weiterlesen

Bei Verwendung des Längencodes 0 wird eine Zahl in die angegebene Zelle gebracht. Der Cursor positioniert hinter die letzte Eingabe, auch beim nächsten Eingabebefehl. Wird die Eingabe mit :RETURN: abgeschlos - sen, wird der nächste Befehl übersprungen.

Beispiel: 4000 5131 5032 1000
nach Ausführung dieser 4 Befehle stehen die eingelesenen Daten in Zelle 31 und 32. Die Eingaben erfolgen hintereinander. Der Befehl 1000 wird nur ausgeführt, wenn die 2. Eingabe mit einem Zwischenraum abschliesst.
Beispiel: 5340 5050 6040
Die Ausgabe erfolgt nicht, wenn die letzte Eingabe mit :RETURN: beendet wurde.

3.3.4 Befehlscode 6 : Ausgeben (aus)

Dem Dialogbenutzer wird die im Längencode des Befehls angegebene Anzahl von Daten aus der im Adresscode angegebenen und aus den folgenden Zellen auf dem Bildschirm in der AUSGABE - Zeile gezeigt.

Mehrere Ausgabebefehle nacheinander erzeugen mehrere Datensätze auf dem Bildschirm, wobei jeder in einer neuen Zeile beginnt. Sie zuletzt ausgegebenen Zeilen werden dabei eine Zeile nach oben verschoben, so daß auf dem Bilschirm jeweils die letzten 3 Ausgabezeilen erhalten bleiben.

Beispiel: 6123 6124 6125
zeigt die Inhalte derq Zellen 23,24,25 untereinander auf dem Bildschirm

6323
zeigt die gleichen Zahlen, jedoch nebeneinander in einer Zeile durch je einen Zwischenraum getrennt.

Speziell: 6100
zeigt den AK – Inhalt.

Sonderfall: 60xx weiterschreiben

Bei Verwendung des Längencodes 0 wird eine Zahl ausgegeben, und zwar hinter die letzte Ausgabe.
Beispiel: 5340 6340 5040 6040
Es werden 4 Zahlen in einer Zeile ausgegeben.

3.3.5 Befehlscode 7 : Vergleichen (vgl)

Der Inhalt des AK wird mit dem Inhalt der im Befehl angegebenen Zellen verglichen. Anschliessend wird bei < der nächste Befehl,
bei = der übernächste Befehl und
bei > der drittnächste Befehl ausgeführt.

Speziell: 7000
überspringt den nächsten Befehl, denn der AK – Inhalt ist gleich dem Inhalt der Zelle 00.

Beispiel: 7031 7000 8025
Nach Ausführung dieser drei Befehle wird der nächste Befehl oder der Befehl aus Zelle 25 genommen, jenachdem ob der AK – Inhalt ungleich bzw. gleich dem Inhalt der Zelle 31 ist.

3.3.6 Befehlscode 8 : Springen (spr)

Als nächster wird der Befehl ausgeführt, dessen Adresse in dem Sprungbefehl angegeben ist. Dabei wird die Adresse des eigentlich nächsten Befehls für spätere Verwendung als Rückkehradresse in Zelle 02 gerettet. Dieses Aufheben erfolgt jedoch nur, falls der Registercode = 0 ist (normalerweise).

Sonderfall: 8000 (Retten des Befehlsregisters)
In diesem Fall wird kein Sprung ausgeführt, sondern nur die Rückkehradresse gerettet.

q

3.3.7 Befehlscode 9 : Holen (hol)

In den AK wird der Inhalt der im Befehl angegebenen Zelle geholt.

Beispiel: 9087
Der Inhalt der Zelle 87 wird in den AK geholt.

Speziell: 9000 Keine Operation
denn Zelle 00 ist der AK.

3.4 Adressrechnung (Indizierung)

Bei allen Befehlen (ausgenommen Ein - Ausgabebefehle) wird eine Adressrechnung durchgeführt, wenn der Registercode (die 2. Ziffer des Befehls) nicht 0 ist. Als Adresse des Operanden wird statt des Adresscodes eine sogenannte effektive Adresse verwendet, die wie folgt berechnet wird:

Effektive Adresse - Adresscode plus Inhalt des Registers, das im Registercode genannt ist.

Beispiel: Vorgegeben: 00:4711,03:0008,29:0013
Befehl: 1321
Ergebnis: 4724

Speziell: 8100 Endlose Wiederholung dieses Befehls
8101 Keine Operation, nächster Befehl
8199 Wiederholung des letzten Befehls
8200 Rücksprung (Unterprogrammende)
8298 Rücksprung zur bedingten Wiederholung des Programms bei Aufruf mit :
70aa 80bb

3.5 Fehlerstop

Bei jedem Fehler in einem Programm sowie bei Ausführung eines Stopbefehls wird die Ausführung beendet. Der Befehlszeiger zeigt dann auf die Zelle, die den Befehl enthält, der den Stop verursacht hat.

Fehlermöglichkeiten:

1. STOP - Befehl (leere Speicherzelle)
2. Speichern oder Einlesen in Zelle 01 (BZ)
3. Division durch Null
4. Ungültiger Befehl, Operationscode keine Ziffer (-) oder ungültige Adressangabe

4. Übungs – Beispiele

BEISP: Das Programm BEISP berechnet

$$5 * 8 + 3 * 7 - 22$$

nach Formel

$$(91)*(92)+(93)*(94)-(95)$$

```
     M O C O - Modellcomputer
     AK   BZ   02   03   04   05   06   07   08   09
 0 0000 0010 0000 0000 0000 0000 0000 0000 0000 0000
10 1091 3092 0096 2096 1093 3094 1096 2095 6100 0000
20 0000 0000 ...
80                                          ... 0000
90 0000 0005 0008 0003 0007 0022 0000 0000 0000 0000
```

WURZEL: Das Programm Wurzel liest eine Zahl ein, berechnet iterativ in einem Unterprogramm die Wurzel und zeigt Näherungswerte.

```
     M O C O - Modellcomputer
     AK   BZ   02   03   04   05   06   07   08   09
 0 0000 0050 0000 0000 0000 0000 0000 0000 0000 0000
10 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
20 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
30 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
40 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
50 5190 6290 9090 8070 7091 7000 8050 0091 8051 0000
60 0000 0010 0000 0000 0000 0000 0000 0000 0000 0000
70 4076 1076 4075 0076 8200 0002 0001 0000 0000 0000
80 0000 0010 0000 0000 0000 0000 0000 0000 0000 0000
90 0000 0010 0000 0000 0000 0000 0000 0000 0000 0000
```

Softwareklasse 2 (Anwendersoftware):

Regelmäßige Wartung

Autoren:

Peter Heyderhoff

Kontaktadresse:

GMD
Postfach 1240

5205 Sankt Augustin 1

Umschlaggestaltung:

Hannelotte Wecken

Druck:

Zentrale Vervielfältigungsstelle der Universität Bielefeld
im Juli 1984

Hinweis:

Diese Dokumentation wurde mit größtmöglicher Sorgfalt erstellt. Dennoch wird für die Korrektheit und Vollständigkeit der gemachten Angaben keine Gewähr übernommen. Bei vermuteten Fehlern der Software oder der Dokumentation bitten wir um baldige Meldung, damit eine Korrektur möglichst rasch erfolgen kann. Anregungen und Kritik sind jederzeit willkommen.