

Matrixmodul

Roland Geißler, Mathias Riesen, Friedhelm Seifert
 VEB Robotron-ZFT, Dresden
 Dr. Rolf Grüner, Klaus Wendlik, VEB Robotron-ZFT, Karl-Marx-Stadt

Für die Zentraleinheiten (ZE) EC 2655 und EC 2655M gibt es ein Spezialrechenwerk (Matrixmodul, kurz MAMO), das als Zusatzeinrichtung angeboten wird. Der MAMO ist nicht an andere Zentraleinheiten des ESER anschließbar.

Die Auswahl der verwirklichten Algorithmen wurde mit der Zielstellung getroffen, einem möglichst breiten Nutzerkreis eine beträchtliche Beschleunigung seiner Aufgabenbearbeitung zu ermöglichen. Wesentliche Anwendungsbereiche sind:

- Seismische Datenverarbeitung
- Optikberechnungen
- Bildverarbeitung
- Energiewirtschaft
- Baustatik
- Optimierungsprobleme der elektronischen Industrie
- Forschungsaufgaben
- Statistik der Volkswirtschaft
- Meteorologie
- Eigenwertprobleme.

Die Leistungsfähigkeit der MAMO-Feidbefehle liegt um den Faktor < 70 über der von entsprechenden Assemblerprogrammen auf EC 2655. Die hohe Parameterflexibilität der MAMO-Feldbefehle gestattet einen effektiven Zuschnitt auf zahlreiche Nutzerprobleme. Neben dem Leistungsaspekt ist für den MAMO-Anwender die starke Vereinfachung seiner Programme durch die Anwendung von MAMO-Befehlen bedeutsam. Insgesamt werden dem Anwender 28 MAMO-Feldbefehle zur Lösung seiner Aufgaben zur Verfügung gestellt.

1. Anschluß und grundsätzliche Betriebsweise des MAMO

Einige Spezialprozessoren, z. B. IBM 2938 und IBM 3838, werden innerhalb der DV-Anlage wie ein E/A-Gerät über das E/A-System betrieben. Untersuchungen ergaben, daß die Einführung von leistungsfähigen Spezial-

funktionen allein nur einen Teil der Nutzerbedürfnisse befriedigt. In der Praxis sind die Spezialfunktionen fast immer in eine Umgebung von Universalprozessorbefehlen bzw. -programmen eingebettet.

Letztlich ist die Ausführungszeit der gesamten Spezialaufgabe für den Nutzer in-

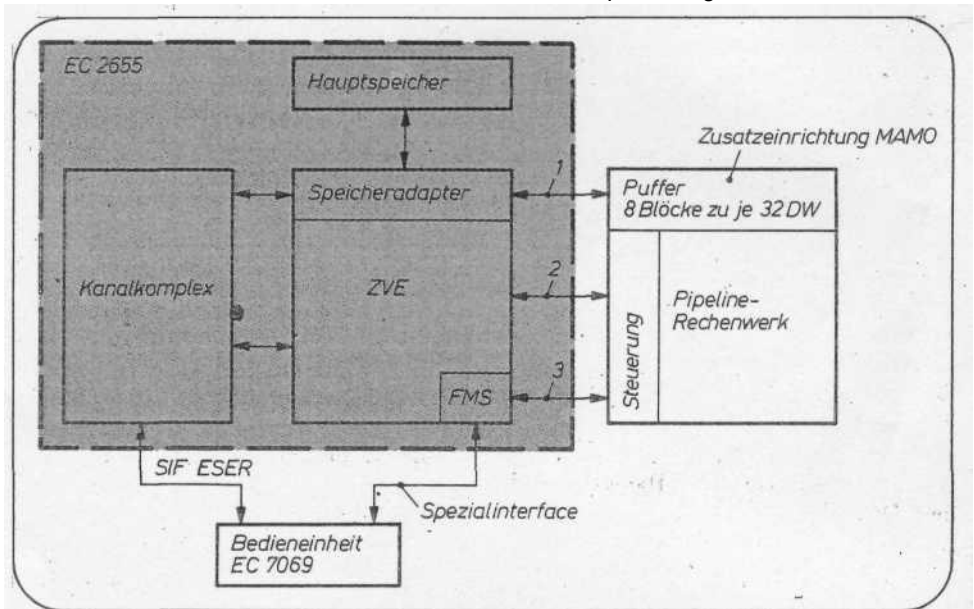


Abb. 1 Anschluß des Matrixmoduls über ein dreiteiliges Spezialinterface

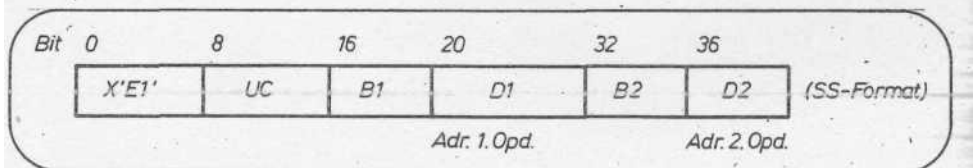


Abb. 2 Struktur eines MAMO-Befehls

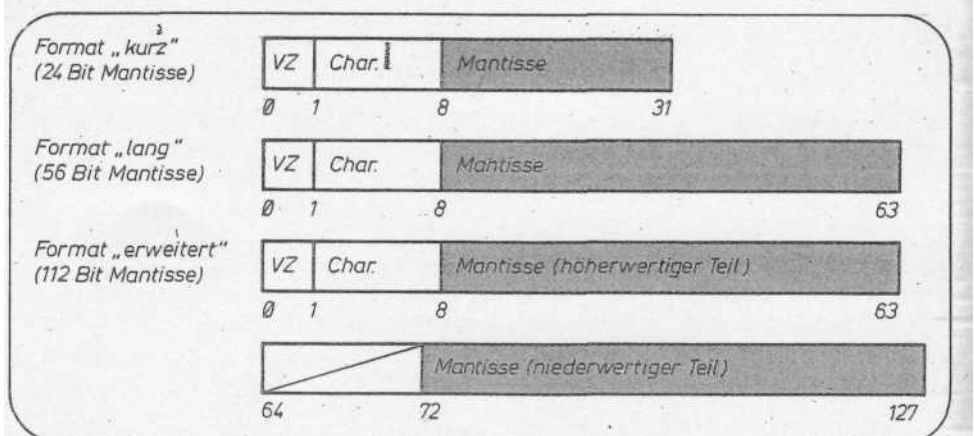


Abb. 3 Gleitkommazahl mit einem Format entsprechend den ESER-Operationsprinzipien

teressapt und nicht die Ausführungszeit einer isolierten Spezialfunktion, d. h., daß zur Lösung einer Aufgabe mit Spezialfunktionen eine gerätetechnische Lösung gefunden werden muß, die den explizit notwendigen Organisationsaufwand (Betriebssystem und eventuelle Monitorprogramme) zur Erledigung der Aufgabe auf ein Minimum reduziert. Dazu gehören aber auch Eigenschaften, die die Arbeit des Programmierers vereinfachen helfen, wie

- Nutzung des virtuellen Speichers
- einfache Einbindungsmöglichkeit der Spezialfunktionen in Programme, die in höheren Programmiersprachen formuliert sind.

Darüber hinaus mußten noch Aspekte der effektiven Weiterentwickelbarkeit der Gerätelösung berücksichtigt werden, die eine sichere Nutzung der Fortschritte auf den Gebieten der Systemarchitektur und der Rechnertechnologie gestatten. Für den vorgesehenen Funktionsumfang und die anvisierten Anwendungsgebiete hat sich die Einbindung der Spezialfunktionen als Erweiterung des Befehlsspektrums des Universalprozessors als effektivste Lösung ergeben. Abb. 1 zeigt, daß über ein dreiteiliges Spezialinterface (1, 2, 3) der MAMO als Rechenwerkserweiterung angeschlossen ist. Der MAMO besteht physisch aus einem ESER-Sehrank einschließlich Stromversorgung.

- über das *Teilinterface 1* erfolgt der Datenaustausch vom MAMO-Datenpuffer über den Speicheradapter mit dem Hauptspeicher (HS). Die Datenwege haben Doppelwortbreite. Im Speicheradapter werden *entweder* ZVE-Zugriffe (ZVE = Zentrale Verarbeitungseinheit) *oder* MAMO-Zugriffe vermittelt. Welche Anforderungen gewertet werden, hängt von einem MAMO-Modustrigger ab, welcher bei Aktivitätenwechsel durch ein ZVE-Mikroprogramm umgeschaltet wird. Die HS-Zugriffe des Kanalkomplexes haben grundsätzlich Priorität

* Das *Teilinterface 2* gestattet einen direkten Steuerdatenaustausch zwischen

dem Rechenwerk der ZVE und dem Steuerkomplex des MAMO über dieses Interface werden Steuerungsfunktionen abgewickelt, die dem Start, der Fortsetzung unterbrochener MAMO-Befehle und der Beendigung der algorithmischen MAMO-Abläufe dienen. Ebenfalls sind die Zugriffsinformationen (Adressen usw.) für die HS-Zugriffe des MAMO darin enthalten.

* Das *Teilinterface 3* dient der Diagnose sowie der Service-Bediensteuerung durch die Bedieneinheit EC 7069 oder den Bedien- und Serviceprozessor (BSP) EC 7069M. Alle drei Teilinterfaces arbeiten synchron mit einem Taktzyklus von 380 ns.

Die ZVE führt für das Zusatzrechenwerk MAMO einige Standardfunktionen aus, z.B.:

- Befehlsvorbereitung und -decodierung
- Start algorithmischer Abläufe im MAMO
- alle Arten der MAMO-Befehlsbeendigung
- Maschinenfehlerauswertung entsprechend den ESER-Operationsprinzipien
- Organisation und übergeordnete Steuerung aller Diagnoseprozesse im MAMO durch ZVE-Mikroprogramme
- Vermittlung der MAMO-HS-Zugriffe
- Kommunikation mit EC 7069 usw.

Darüber hinaus werden alle MAMO-Steuerbefehle von der ZVE ausgeführt sowie alle Betriebszustände des MAMO überwacht.

Das Pipeline-Rechenwerk im MAMO ist als Hochleistungsrechenwerk in der Lage, bei kurzem Format in der Verarbeitung (kurze ESER-Gleitkommazahlen) gleichzeitig 2,5 Millionen Additionen und 2,5 Millionen Multiplikationen je Sekunde auszuführen.

2. Allgemeine MAMO-spezifische Grundbegriffe

2.1. MAMO-Befehle

Der MAMO wird durch Maschinenbefehle angesprochen. Sie stellen die Erweiterung des Befehlsspektrums der EC 2655 dar. MAMO-Programme bestehen aus einer Folge von ZVE- und MAMO-Befehlen, die alle unter der Steuerung des gleichen PSW laufen. Alle MAMO-Befehle (Abb. 2) haben den Operationscode 'X'EI'. Sie werden durch den Untercode (UC) unterteilt in:

- Steuerbefehle (UC = 00... 7F)
- Feldbefehle (UC = 80... FF).

Die neun Steuerbefehle lassen sich nicht unterbrechen. Sie sind zum Teil privilegiert. Sie werden bis auf einen Steuerbefehl in der ZVE allein realisiert und benutzen den MAMO nicht. Die Feldbefehle sind unterbrechbar. Sie werden gemeinsam von der ZVE und vom MAMO verwirklicht. Der MAMO übernimmt dabei im wesentlichen den algorithmischen Teil der Befehle. Die Feldbefehle verarbeiten im allgemeinen große Datenmengen, die nicht im Befehl selbst beschrieben werden können. Daher verweist die Adresse des zweiten Operanden (Basisregister B2, Verschiebung D2) auf den *Parameterblock (PAB)*, der alle erforderlichen Angaben über die zu verarbeitenden und zu erzeugenden Daten sowie weitere Parameter zur Steuerung der MAMO-internen Abläufe enthält. Die Adresse des ersten Operanden verweist auf den *Mitteilungsblock (MIB)*, in dem der MAMO bei Operationsende alle notwendigen Informationen für den Nutzer, z. B. Mitteilungen über die "erzeugten Daten bzw. bei Abbruch des algorithmischen Teils, Angaben zur Ursache, zur Verfügung stellt.

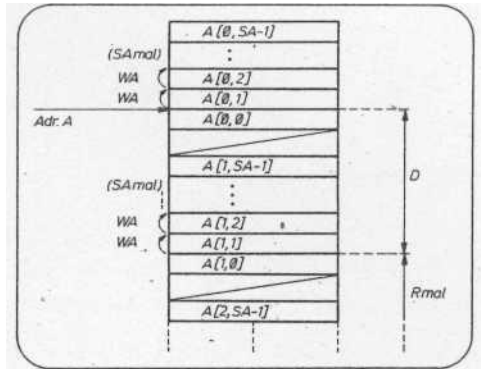


Abb. 4 Beispiel für ein nichtindiziertes Feld mit $WA < 0$

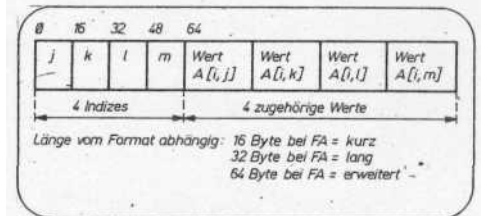


Abb. 5. Komponentenblock

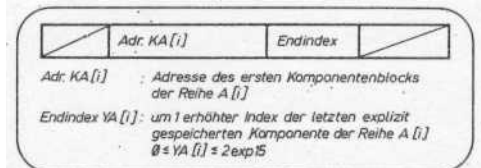


Abb. 6 Reihenkopf RK

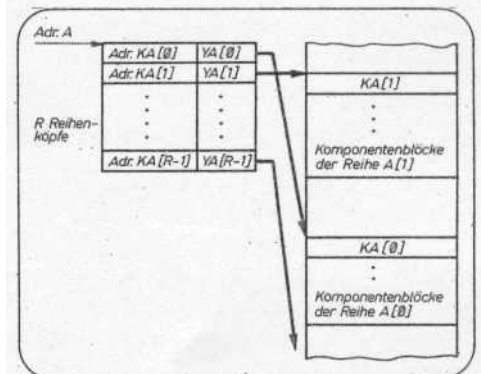


Abb. 7 Beispiel eines indizierten Feldes

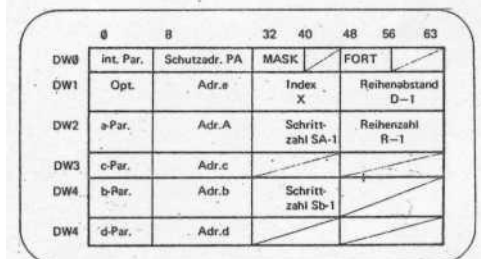


Abb. 8 Struktur des Parameterblocks PAB

2.2. Datenarten und Datentypen

In MAMO-Feldbefehlen werden Indizes, Komponenten, Reihen, Felder und Skalare verarbeitet.

Ein *Index j* ist eine vorzeichenlose Festkommazahl im Bereich $0 \leq j \leq 2 \exp 15 - 1$.

VZT	Verarbeitung bzw. Ausgabe
00	Wert
01	- Wert
10	Wert
11	- Wert

Abb. 9 Vorzeichentransformation

Eine Komponente $A[i, j]$ bzw. $a[j]$ ist ein Wert (Abb. 3), dem ein Index j zugeordnet werden kann, dabei liegt die *Reihennummer* i im Bereich $0 \leq i \leq 2 \exp 15 - 1$.

Eine Reihe a bzw. $A[i]$ ist eine geordnete Menge von Komponenten gleichen Formats (z. B. Vektor); die Anzahl der Komponenten heißt *Schrittzahl* S , $0 \leq SA - 1 \leq 2 \exp 15 - 1$.

Ein Feld A ist eine geordnete Menge von Reihen gleicher Schrittzahl gleichen Formats (z. B. Matrix); die Anzahl der Reihen heißt *Reihenzahl* R , $0 \leq R - 1 < 2 \exp 15 - 1$.

Ein Skalar d ist ein Wert, dem kein Index zugeordnet werden kann.

Die Speicherung von Komponenten, Reihen und Feldern kann auf zwei verschiedene Arten geschehen:

– *Nichtindizierter Typ* der Datenspeicherung, wo nur die Werte abgespeichert werden

– *indizierter Typ* der Datenspeicherung, wo zu allen Werten der Komponenten die zugehörigen Indizes explizit mit abgespeichert werden (in der Literatur vielfach als kompakte Speicherung bezeichnet).

2.2.1. Nichtindizierte Komponenten, Reihen und Felder

Eine *nichtindizierte Komponente* $A[i, j]$ besteht aus einem Wert vom Format der Komponenten; der Index wird nicht gespeichert, er wird impliziert durch die Relativadresse des Speicherplatzes vom Wert, bezogen auf den Reihenanfang, bestimmt.

Eine *nichtindizierte Reihe* $A[i]$ besteht aus den nichtindizierten Komponenten $A[i, 0], A[i, 1], \dots, A[i, SA - 1]$; dabei wird die Adresse des Wertes der Komponenten $A[i, 0]$ als *Adresse der Reihe* $Adr. A[i]$ bezeichnet. Die Wertadressen der einzelnen Komponenten der Reihen werden bestimmt durch:

$Adr. A[i, j]$ in Byte: $= Adr. A[i] + j * WA * F$

$j = 0, 1, \dots, SA - 1$

WA : Schrittwerte, $WA = \pm 1, 2, 4, 8$

$F = \begin{cases} 4 & \text{bei FA} = \text{kurz} \\ 8 & \text{bei FA} = \text{lang} \\ 16 & \text{bei FA} = \text{erweitert} \end{cases}$

Bei Resultatreihen ist die Schrittwerte stets $W_c = \pm 1$, d. h., die nichtindizierte Reihe c entsteht durch lückenloses Abspeichern der Werte der Komponenten.

Ein *nichtindiziertes Feld* A (Abb. 4) besteht aus nichtindizierten Reihen $A[0], A[1], \dots, A[R - 1]$ gleicher Schrittwerte; die Adresse der Reihe $A[0]$ wird dabei als *Adresse des Feldes* $Adr. A$ bezeichnet.

Die Adressen der Reihen des Feldes werden bestimmt durch:

$Adr. A[i]$ in Byte: $= Adr. A + i * D * F$

$i = 0, 1, \dots, R - 1$

D : Reihenabstand, $0 \leq D - 1 \leq 2 \exp 15 - 1$.

2.2.2. Indizierte Komponenten, Reihen und Felder

Eine *indizierte Komponente* $A[i, j]$ besteht aus einem Wert vom Format der Komponente und dem explizit gespeicherten Index j . Komponenten, deren Wert eine echte Null ist, müssen nicht abgespeichert werden. Es werden jeweils vier Komponenten, deren Wert und Index explizit gespeichert sind, zu einem *Komponentenblock* zusammengefaßt (Abb. 5).

Abb. 10 Substitution entsprechend einer Schranke

$e \leq 0$		$e > 0$	
Substitutionsbedingung	Substitution	Substitutionsbedingung	Substitution
0	keine	$ w < e$	$w := 0$
$w < e$	$w := e$	$W < e$	$w := 0$
1	$w := 0$	$ w > e$	$w := \text{sign} w * e$
$w > e$	$w := 0$	$w > e$	$w := e$

Abb. 11 Mitteilungsblock MIB

	0	8	32	48	63
DW0	Adr. des Feldbefehls				
DW1	Folgeadresse		Zahl signif. Komponenten		
DW2	Relativadr. des Extremwertes		Extremwertindex		
DW3	Relativadr. des Wertes gem. Option 4		Wertindex		

Abb. 12 Algorithmische Ausnahmebedingungen

Ausnahmart	CC	Erläuterung
IP	2	unzulässige Parameterangabe im PAB in einer indizierten Reihe bilden die Indizes keine monoton steigende Folge
NM	2	
SO	1	es wurde versucht, eine Komponente der Resultatreihe auf die Adresse zu schreiben, die durch die Schutzadr. PA im PAB angegeben wurde
EOA	2	Exponentenüberlauf bei Addition, der Mantisse = 0, Charakteristik ≥ 128
EUA	1	Exponentenüberlauf bei Addition, d. h., Mantisse = 0, Charakteristik < 0
LS	1	Mantissenverlust bei Addition, d. h., Mantisse = 0

Abb. 13 Art der algorithmischen Ausnahmebedingungen

	0	8	24	32	48	63
DW0	Adr. des Feldbefehls					
DW1	Art der Ausnahme		Quelle der Ausnahme		Index	Reihen-Nr.

Eine *indizierte Reihe* $A[i]$ entsteht aus indizierten Komponenten durch lückenloses Aneinanderreihen von Komponentenblöcken. Der Zugriff zu einer indizierten Reihe erfolgt über den *Reihenkopf* RK (Abb. 6).

Enthält eine Reihe $A[i]$ keine explizit gespeicherte Komponente, d. h., alle Werte der Reihe sind Null, so wird $YA[i] = 0$ gesetzt.

Ein *indiziertes Feld* A (Abb. 7) ergibt sich durch Aneinanderreihen der Reihenköpfe der indizierten Reihen $A[0], A[1], \dots, A[R - 1]$, wobei die Adresse der Reihe $A[0]$, d. h. des ersten Reihenkopfes, als Adresse des Feldes $Adr. A$ bezeichnet wird. Die Adressen der einzelnen Reihen, d. h. der Reihenköpfe, des Feldes werden bestimmt durch:

$Adr. A[i]$ in Byte: $= Adr. A + i * 8$

$i = 0, 1, \dots, R - 1$

2.3. Parameterblock PAB

Der Parameterblock PAB umfaßt die erforderlichen Angaben über die Daten-

Mnemonic	Bezeichnung	Untercode	Kurzbeschreibung	Mnemonic	Bezeichnung	Untercode	Kurzbeschreibung
Reihenmanipulationen				Feldoperationen			
TER	Testen Reihe (Test row)	85	Testen der Komponenten $a[X], a[X+1], \dots, a[SA-1]$	SMARR	Bilden Skalarprodukt der Reihen eines Feldes mit einer Reihe (Scalar multiply array rows by row)	C2	$c[j] := \sum_{i=\emptyset}^{\min(SA, Sb)-1} A[j,i] * b[i]$ ($j = \emptyset, 1, \dots, R-1$)
RSTR	Umspeichern Reihe (Restore row)	A1	$c[j] := a[j+X]$ ($j = \emptyset, 1, \dots, SA-1-X$)	SMARI	Bilden Skalarprodukt der Reihen eines Feldes mit sich selbst (Scalar multiply array rows by itself)	A6	$c[j] := \sum_{i=\emptyset}^{SA-1} A[j,i] * abs(A[j,i])$ ($j = \emptyset, 1, \dots, R-1$)
URR	Vereinigen Reihe mit Reihe (Unite row and row)	C1	$c[j] := \begin{cases} b[j] & \text{für } j < X \\ a[j-X] & \text{für } j \geq X \end{cases}$ ($j = \emptyset, 1, \dots, SA-1+X$)	ARR	Addieren Reihe und Reihe (Add row and row)	D7	$c[j] := a[j] + b[j]$ ($j = \emptyset, 1, \dots, \max(SA, Sb)-1$)
ARC	Addieren Reihe und Konstante (Add row and constant)	DE	$c[j] := a[j] + b[\emptyset]$ ($j = \emptyset, 1, \dots, SA-1$)	SRVAR	Summieren Werte einer Reihe für alle Reihen eines Feldes (Sum row values for array rows)	B2	$c[j] := \sum_{i=0}^{SA-1} A[j,i]$ ($j = \emptyset, 1, \dots, R-1$)
MRS	Multiplizieren Reihe mit Skalar (Multiply row by scalar)	C4	$c[j] := d * a[j]$ ($j = \emptyset, 1, \dots, SA-1$)	LCARRC	Berechnen Linearkombination der Reihen eines Feldes gemäß Reihenkombination (Linear combine array rows according to row components)	E8	$c[j] := \sum_{i=\emptyset}^{\min(R, Sb)-1} A[j,i] * b[i]$ ($j = \emptyset, 1, \dots, SA-1$)
MRSAR	Multiplizieren Reihe mit Skalar und Addieren Reihe (Multiply row by scalar and add row)	E7	$c[j] := d * a[j] + b[j]$ ($j = \emptyset, 1, \dots, \max(SA, Sb)-1$)	SAR	Summieren Reihen eines Feldes (Sum array rows)	C8	$c[j] := \sum_{i=\emptyset}^{R-1} A[i,j]$ ($j = \emptyset, 1, \dots, SA-1$)
MRSARX	Multiplizieren Reihe mit Skalar und Addieren Reihe ab Index X (Multiply row by scalar and add row from index X)	F7	$c[j] := \begin{cases} b[j] & \text{für } j < X \\ d * a[j] + b[j] & \text{für } j \geq X \end{cases}$ ($j = \emptyset, 1, \dots, \max(SA, Sb)-1$)	SAVD	Summieren Werte in der Hauptdiagonalen (Sum values in principal diagonal)	BA	$c[\emptyset] := \sum_{i=\emptyset}^{\min(R, SA)-1} A[i,i]$
MRR	Multiplizieren Reihe mit Reihe (Multiply row by row)	C7	$c[j] := a[j] * b[j]$ ($j = \emptyset, 1, \dots, SA-1$)	CPRV	Berechnen Polynom für Werte einer Reihe (Calculate polynomial for row values)	CE	$c[j] := \sum_{i=\emptyset}^{Sb-1} b[i] * a[j] \exp(Sb-1-i)$ ($j = \emptyset, 1, \dots, SA-1$)
MRI	Multiplizieren Reihe mit sich selbst (Multiply row by itself)	A7	$c[j] := a[j] * abs(a[j])$ ($j = \emptyset, 1, \dots, SA-1$)				
Feldmanipulationen				Korrelation			
FRACI	Bilden Reihe aus Feldkomponenten mit gegebenem Index (Form row from array components with given index)	A3	$c[j] := A[j, SA-1]$ ($j = \emptyset, 1, \dots, R-1$)	COR	Korrelation (Correlation)	CF	$c[j] := \sum_{i=\emptyset}^{\min(SA-j, Sb)-1} b[i] * a[j+i]$ ($j = \emptyset, 1, \dots, SA-1$)
FRACD	Bilden Reihe aus Feldkomponenten in der Hauptdiagonalen (Form row from array components in principal diagonal)	AA	$c[j] := A[j,j]$ ($j = \emptyset, 1, \dots, \min(R, SA)-1$)	AC	Autokorrelation (Autocorrelation)	AF	$c[j] := \sum_{i=\emptyset}^{SA-j-1} a[i] * a[j+i]$ ($j = \emptyset, 1, \dots, SA-1$)
EARRVI	Vertauschen Werte der Komponente mit gegebenem Index in Feldreihen (Exchange in array rows values according to indices)	A9	$A[j,i] := \begin{cases} A[j,i] & \text{für } i \neq X, \\ & i \neq SA-1 \\ A[j,X] & \text{für } i = SA-1 \\ A[j,SA-1] & \text{für } i = X \end{cases}$ ($j = \emptyset, 1, \dots, R-1$)	CCR	Zyklische Korrelation (Cyclic correlation)	DF	$c[j] := \sum_{i=\emptyset}^{s-1} b[i] * a[j+i] \text{ wenn } s \leq SA-j$ $+ \sum_{i=SA-j}^{SA-j-1} b[i] * a[j+i-Sa]$ wenn $s > SA-j$ ($j = \emptyset, 1, \dots, SA-1$), $s = \min(SA, Sb)$
ZAVI	Löschen Werte von Feldkomponenten mit gegebenem Index (Zero array values with given index)	89	$A[j,i] := \begin{cases} A[j,i] & \text{für } i \neq SA-1 \\ \emptyset & \text{für } i = SA-1 \end{cases}$ ($j = \emptyset, 1, \dots, R-1$)				

Mnemonic	Bezeichnung	Untercode	Kurzbeschreibung
FFT, Komplexe Operationen			
FFT	Schnelle Fouriertransformation (Fast Fourier Transformation)	CD	Schnelle Fouriertransformation der Reihe a in die Reihe c ($Sa = 2 \exp m$ Kompl. Komponenten) nach dem Algorithmus von COOLEY-TUKEY
FFTI	Inverse schnelle Fouriertransform. (Fast Fourier Transformation invers)	DD	
RAFC	Umordnen Fourierkoeffizienten (Rearrange Fourier coefficients)	CC	Umordnen der bei FFT/FFTI entstandenen Reihe a in die Reihe c, Multiplizieren mit Skalar d ($Sa = 2 \exp m$ komplexe Komp.)
MRRC	Multiplizieren Reihe mit Reihe, Komplex (Multiply row by row, complex)	C6	$c[j] := a[j] * b[j]$ (komplex) ($j = 0, 1, \dots, Sa-1; Sa = k * 32$ Kompl. Komp.)

Abb. 14
Liste der MAMO-Feldbefehle (Fortsetzung von S. 35)

Struktur sowie die zur Verarbeitung der Daten im MAMO (Abb. 8). Abhängig vom jeweiligen Feldbefehl sind verschiedene Doppelworte (DW) nicht definiert, z. B. für Skalar d DW5, für Operand b DW4 u. ä. Außer den Angaben für die Datenstruktur, wie Adressen Adr. A, b, c, d, Schrittzahlen SA — 1, Sb — 1, Reihenzahl R-I, sind im DW2-5 des PAB noch die Parameter Schrittweite WA, Wb, Wc, TypTYA, TYb, TYc sowie Format FA, Fb, Fc, Fd enthalten. Als weiterer Parameter kann die *Vorzeichen-transformation* VZTA, VZTb, VZTc, VZTd ausgewählt werden. Die Vorzeichen-transformation (Abb. 9) bietet die Möglichkeit, die Komponenten einer Reihe, eines Feldes bzw. eines Skalars mit einem anderen Vorzeichen als vorgegeben im MAMO zu verarbeiten bzw. mit einem anderen Vorzeichen als berechnet zu speichern. Durch die im DW1 angegebenen *Optionen* kann die Abarbeitung von MAMO-Befehlen modifiziert bzw. um zusätzliche Funktionen erweitert werden: *Option 1* — Rücksetzen der Schreibadresse, *Option 2* — Testen der Resultatkomponenten, *Option 3* — Speichern expliziter Nullen, *Option 4* — Suchen einer Resultatkomponente.

- *Option 1* ist wirksam, wenn versucht wird, auf die im DW0 angegebene *Schutzadresse PA* in den HS zu speichern; wodurch die algorithmische Ausnahmebedingung SO verursacht wird. Bei gewählter Option 1 wird beim Fortsetzen des unterbrochenen Feldbefehls das Doppelwort, dessen Einschreiben in den HS durch SO unterbrochen wurde, auf die ursprüngliche Adr. c bzw. Adr. Kc geschrieben. Die weiteren Komponenten der Resultatreihe werden fort-

laufend eingeschrieben. Vom Nutzer muß vor der Fortsetzung des Feldbefehls der Speicherbereich im HS gerettet werden. • *Option 2* ermöglicht:

- Substitution gemäß Schranke
- Extremwertsuche.

Bei *Substitution gemäß Schranke e* (Adresse Adr. e in DW1) werden die Werte W der Resultatkomponenten, die die Substitutionsbedingung erfüllen, durch eine echte Null gleichen Formats bzw. der Betrag des Wertes wird durch den Betrag der Schranke ersetzt (Abb. 10).

Bei *Extremwertsuche* wird die Komponente der Resultatreihe c [j] gesucht, deren Wert die Suchbedingung erfüllt.

Als Suchbedingungen sind möglich:

$$\begin{aligned} |w| &= \min | \text{Wert } c [j] | \\ w &= \min (\text{Wert } c [j]) \\ |w| &= \max | \text{Wert } c [j] | \\ w &= \max (\text{Wert } c [j]) \end{aligned}$$

- *Option 3* ist nur wirksam, wenn eine indizierte Resultatreihe erzeugt wird. Bei gewählter Option 3 werden die Resultatkomponenten, deren Werte explizit zu berechnen sind, auch explizit gespeichert, (d. h. auch Komponenten mit dem Wert Null).

- Bei *Option 4* wird diejenige Komponente c [X] der Resultatreihe gesucht, die dem *Index X* (in DW1 des PAB enthalten) zugeordnet ist.

Im DW0 des PAB können interne Parameter für die Verarbeitung der Operanden sowie *Maskenbit MASK* und *Fortsetzungsbit FORT* für die Behandlung von algorithmischen Ausnahmebedingungen angegeben werden.

2.4. Mitteilungsblock MIB

Im Mitteilungsblock MIB werden dem Nutzer nähere Informationen bei regulärem Operationsende (Bedingungscode

CC=0) bzw. bei irregulärer Beendigung (CC=1,2) übergeben. Struktur des MIB bei CC=0 (Abb. 11). In DW0 wird die *Adresse* desjenigen *Feldbefehls* angegeben, auf den sich der MIB bezieht. Die in DW1 angegebene *Folgeadresse* zeigt auf die nächste freie Doppelwortadresse nach der Resultatreihe c. Die *Zahl signifikanter Komponenten* enthält bei indizierter Resultatreihe c die Zahl der explizit gespeicherten Komponenten bzw. bei nichtindizierter Resultatreihe c die Zahl der Komponenten von c, deren JWert ungleich Null ist. In DW2 sind Angaben für Option 2 und in DW 3 die für Option 4 enthalten. Die angegebenen *Relativadressen* beziehen sich im allgemeinen auf die Adr. c der Resultatreihe. Zu einer irregulären Beendigung eines Feldbefehls kommt es durch das Auftreten einer algorithmischen Ausnahmebedingung (Abb. 12), wobei mit der Meldung CC=1 eine Fortsetzung möglich ist. Durch die im PAB enthaltenen Fortsetzungsbit FORT kann angegeben werden, ob Fortsetzungsabsicht besteht. Bei der Meldung CC=2 ist eine Fortsetzung nicht möglich. Mit den im PAB enthaltenen Maskenbit MASK kann die Meldung bestimmter algorithmischer Ausnahmebedingungen unterdrückt werden. Abhängig von der aufgetretenen Ausnahme erfolgt eine Eintragung in den MIB. Im DW1 (Abb.13) werden Angaben zur Art der algorithmischen Ausnahmebedingung sowie Hinweise zur Umgebung (Index, Reihen-Nr.) gegeben, durch die die Ausnahme hervorgerufen wurde.

3. Befehlsspektrum

3.1. Liste der MAMO-Befehle

In dieser Liste (Abb. 14) sind die definierten Feldbefehle des Operationscodes

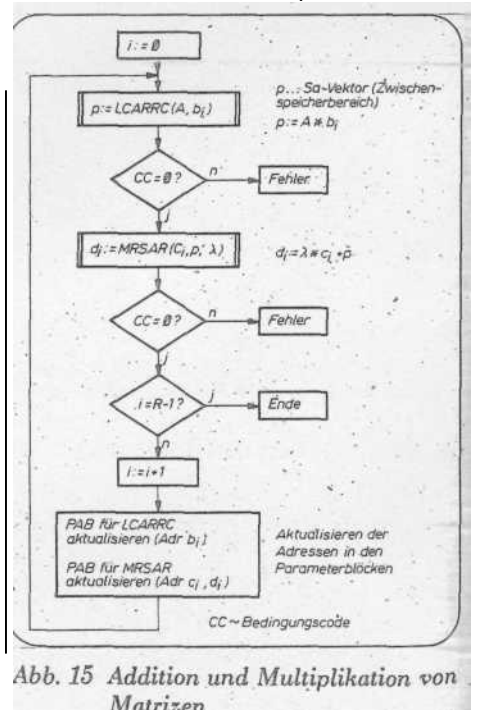


Abb. 15 Addition und Multiplikation von Matrizen

X'EI' aufgeführt. Eine Erweiterung der Befehlsliste ist im Rahmen der realisierten Gerätetechnik durch die Flexibilität der Mikroprogrammsteuerung bedingt möglich.

3.2. Einfache Demonstrationsbeispiele

Beispiel 1 (Abb. 15): Addition und Multiplikation von Matrizen

$D = A * B + \lambda * C$
 $A \dots (S_a, S_b)$ -Matrix
 $B \dots (S_b, R)$ -Matrix,
 $b_i \dots i$ -te-Spalte von B
 $C \dots (S_a, R)$ -Matrix,
 $c_i \dots i$ -te-Spalte von C
 $D \dots (S_a, R)$ -Matrix,
 $d_i \dots i$ -te-Spalte von D
 $(i = 0, 1, \dots, R - 1)$
 $\lambda \dots$ skalarer Faktor

A, B, C, D — spaltenweise gespeichert
Beispiel 2 (Abb. 16): Iteration $X^{(k+1)} = A * X^{(k)}, X^{(0)} = X$;
 Abbruch, wenn $\max |X_i^{(k+1)} - X_i^{(k)}| < e$
 $(e \dots$ Schranke)

$A \dots (R, R)$ -Matrix
 $X, X^{(k)} \dots R$ -Vektoren $(k = 0, 1, \dots)$
 A — zeilenweise gespeichert

Beispiel 3 (Abb. 17): Faltung zweier Reihen a, b mittels schneller Fourier-Transformation (FFT)

$c = f(a, b)$ a, b, c — Reihen mit $S_a = S_b = S_c = 2^m$ Komponenten.

In den Programmablaufplänen zu den Beispielen sind MAMO-Befehle durch \square gekennzeichnet. Andere Aktivitäten werden über ZVE-Befehle verwirklicht.

3.3. MAMO-Befehlszeiten und Leistungsvergleich

Aufgrund der mannigfaltigen Parametereinflüsse auf MAMO-Befehlszeiten ist eine absolute Leistungsangabe unzumutbar. Untersuchungen haben ergeben, daß die Leistungsfähigkeit des MAMO ab Reihenlängen von etwa 30 zum Tragen kommt. In Abb. 18 sind einige Befehlszeiten und Leistungsvergleiche zu entsprechenden Assemblerprogrammen (gerechnet auf der ZE 2655) aufgeführt. Dabei werden Beispiele gewählt, die in der Anwendung wesentliche Bedeutung haben werden.

4. Globale Charakteristik des MAMO

4.1. Blockstruktur

In Abb. 19 ist der MAMO in seine wichtigsten Logikkomplexe untergliedert. Es gibt diese Gruppen von Komplexen:
Zentraler Diagnosekomplex (ZDK): Der ZDK steht mit dem Fehlermaßnahmesystem FMS der ZVE in direkter Verbindung, arbeitet aber zeitweilig völlig autonom. Im Teilinterface 3 ist ein spezieller Kommandoteil enthalten, wo dem ZDK von der ZVE und von EC 7069M/EC 7069 elementare Steuerkommandos übergeben werden, über

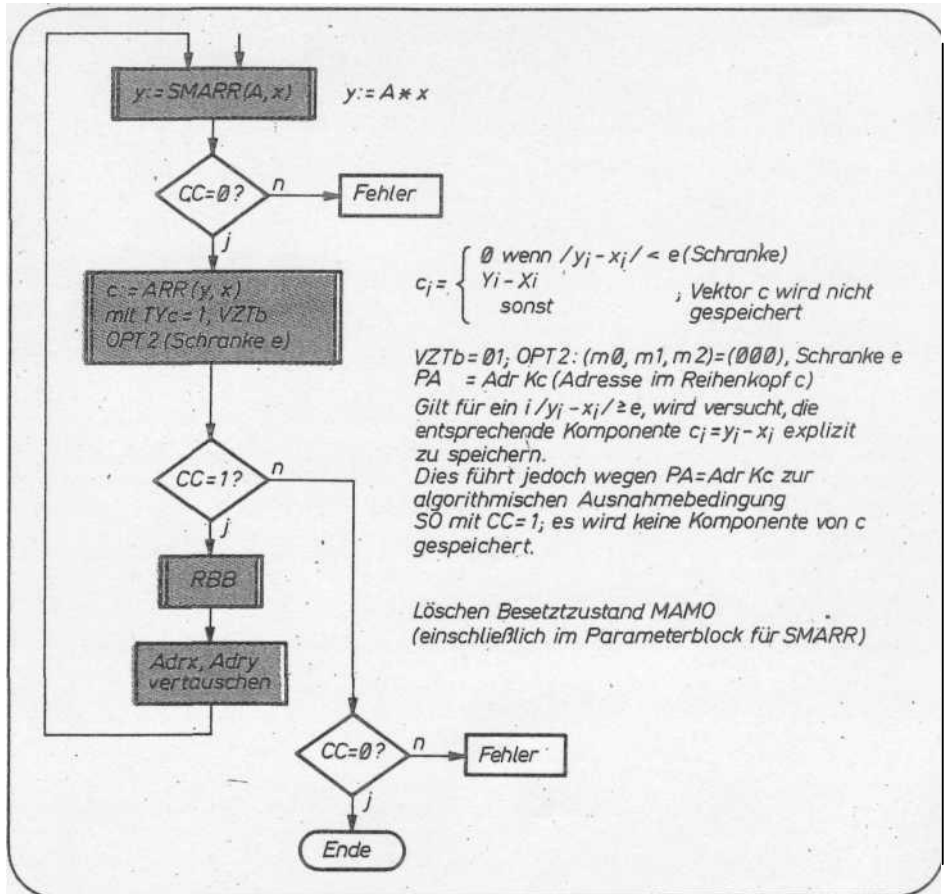


Abb. 16 Iteration mit Abbruch

diese Verbindung kann der ZDK im Rahmen seiner breiten Funktionspalette von außen vollständig programmiert werden. Diese Grundeigenschaft in Verbindung mit der neuartigen orthogonalen Daten-Schiffeneinrichtung „Diagnosebus“ ist Voraussetzung für die außergewöhnlich guten Servicebedien- und Diagnoseeigenschaften des MAMO. Der ZDK nimmt im Sinne der MAMO-Algorithmen keine Verarbeitung vor.

Mikroprogrammgesteuerte Funktionskomplexe:

- MG: MG ist das Mikroprogrammsteuerwerk. Mit seiner Hilfe werden folgende Funktionen ausgeführt:
 - Steuerung der Abläufe im MH zum Erzeugen der notwendigen Adreßrechnungen für alle HS-Zugriffe des Puffers und für spezielle Steuerungszugriffe (dynamisches Mikroprogramm laden)
 - Steuerung der Interfaceprozeduren von MF
 - Voreinstellung und Start der folgegesteuerten Automaten
 - Auswertung der laufenden Betriebszustände der arbeitenden Automaten

- MF: Realisieren der Prozeduren auf Teilinterface 2
- MH: Adreßrechenwerk für den Verkehr des MAMO mit dem HS.

Folgegesteuerter Komplex: Dieser besteht aus sechs größeren Logikeinheiten, die miteinander eine Art „hand shake“-Betrieb ausführen. MQ, MM, MS und MT bilden die Verarbeitungspipeline, welche vom Steuerwerk MI gesteuert wird.

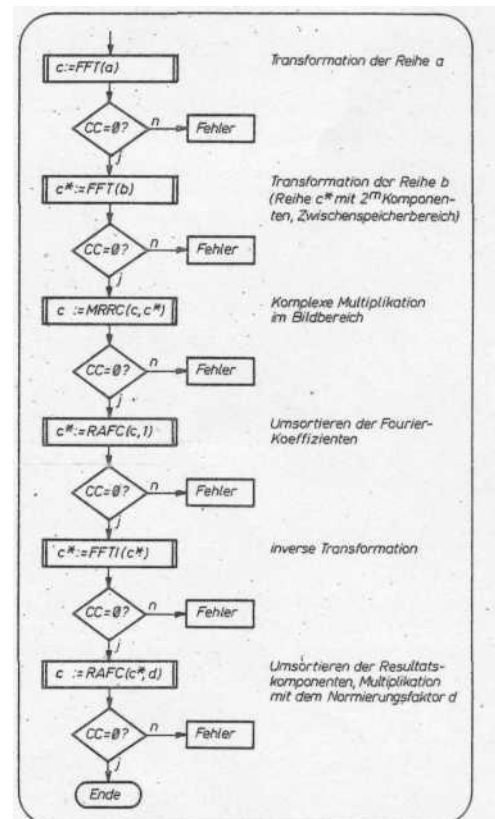


Abb. 17 Faltung zweier Reihen a, b mit der schnellen Fourier-Transformation (FFT)

- MP: Der Puffer besteht aus acht Blöcken von je 32 Doppelworten. Die Blöcke werden dynamisch wechselnd als Ein-/Ausgabepuffer verwendet und arbeiten weitgehend parallel. Vor allen

Befehl	wesentliche Parameter		Befehlszeit t _M /ms	Faktor t _{ZE} /t _M		
	Sa	Sb			Format Faktoren Summ.	
ARR	64	64	—	K	0,15	7
ARR	512	256	—	K	0,6	17
MRSARX	512	384	K	L	0,7	29
m. Extrem- wertsuche						
MRSARX	512	384	L	L	1,2	12
SMARR	64	64	K	K	0,17	9
einreihig						
SMARR	512	512	K	L	0,5	20
einreihig						
CPRV	512	16	K	L	4,0	35
COR	4096	4096	K	K	4100	40
FFT	64	K Komplex			1,2	35
FFT	2048	K Komplex			40	65

t_M: MAMO-Befehlszeit
t_{ZE}: Zeit eines adäquaten Assemblerprogramms, gerechnet auf ZE 2655
Sa/Sb: Schrittzahlen der Reihen a/b
K/L: Formate kurz/lang

Abb. 18
Befehlszeiten und
Leistungsvergleich
zu Assembler-
programmen (ge-
rechnet auf der
ZE EC 2655)

Dingen sind die Versorgungszugriffe der Pipeline und der HS-Verkehr parallelisiert.

- MQ: Datenauswahlsteuerung der Pipeline aus dem Puffer
- MM: Multiplikationswerk
- MS: Additionseinheit
- MT: Testeinheit
- MI: Indexsteuerwerk, welches den Pipelineautomaten die Steuertupel liefert.

In Abb. 20 sind die Steuerungsniveaus in der EC 2655/EC 2655M stilisiert dargestellt. Es ist ersichtlich, daß zum Abarbeiten von MAMO-Aufgaben vier Steuer-niveaus aktiv werden, wobei zum Teil eine Parallelisierung der Funktionen erhebliche Leistungsgewinne bringt.

4.2. Ablauf eines MAMO-Feldbefehls aus Nutzersicht

In Abb. 21 wurde der Versuch unternommen, die wichtigsten Aktivitäten beim Ablauf eines MAMO-Feldbefehls in einem Blockdiagramm darzustellen.

- **Block A:** Kennzeichnet ein ZVE-Programm, das bis zu diesem Zeitpunkt noch keinen MAMO-Feldbefehl abgearbeitet hat.
- **Block B:** Stellt mit der Decodierung des EI-Operationscodes den Beginn des MAMO-Feldbefehls dar. Diese Funktion erfolgt in der Befehlsvorbereitungseinheit der ZVE.
- **Der Block C** repräsentiert ein ZVE-Mikroprogramm, das durch den EI-Befehlscode aktiviert wird. Seine Aufgaben sind:

- Prüfen der Untercodegültigkeit
- eventuelles Erzeugen der Standardprogrammannahmen PRAN (1)
- Prüfen der MAMO-Zustände BE REIT, BESETZT, GESPERRT, Einschaltung MAMO-Modus
- Übergabe des gültigen Untercodes an den MAMO und MAMO-Start. Dieses ZVE-Mikroprogramm (C) geht anschließend in einen speziellen Wartezustand über, in dem es auf eine Mitteilung des MAMO wartet.

- Im **Block D** wird vom MAMO die Abarbeitung des gewünschten Algorithmus vorgenommen. Treten während der Ausführung des Algorithmus durch den MAMO in der ZVE E/A-Unterbrechungen, externe Unterbrechungen usw. auf, die eine unverzügliche Behandlung durch die ZVE erfordern, dann wird von der ZVE-Steuerung dem MAMO eine Halтанforderung (2) übermittelt.

® Daraufhin beendet der MAMO 'seine HS-Zugriffsaktivitäten, geht in den Be-setztzustand über und meldet sich mit einer Haltmitteilung (3) bei einem ZVE-Mikroprogramm, das durch den **Block E** dargestellt wird. Dieses Mikroprogramm leitet ein Operationsende des MAMO-Befehls ein. Anschließend erfolgt die Unterbrechungsbehandlung in der ZVE über die Aktivität 4. Die Aktivität 3 tritt ebenfalls auf, wenn bei der Abarbeitung des Algorithmus algorithmische Ausnahmen oder bei der Datenbeschaffung durch den MAMO im HS Zugriffsausnahmen entstehen.

- Liegt keine Halтанforderung vor bzw. liegt bei Beendigung des ZVE-Mikroprogramms **Block E** keine Unterbrechung in der ZVE vor, dann wird der nächste ZVE-Befehl des aktiven MAMO-Programms aufgerufen, welcher ein bedingter Sprungbefehl sein muß (**Block F**).
- Bei CC = 0 wird das Programm beim Folgebefehl, der bereits wieder ein MAMO-Befehl sein kann, fortgesetzt (**Block G**).
- Beim CC 4= 0 erfolgt der Sprung zu einer Anschlußroutine (**Block I**), die beim Erstaufruf der Makroanweisung MPB (siehe Abschnitt 6.1.) zusammen mit dem Parameterblock PAB erzeugt wird.
- Die Anschlußroutine stellt die Verbindung zur PRAN-Behandlungsroutine des Nutzers her (**Block H**). Die Adresse der Behandlungsroutine steht im Parameterblock PAB. Durch dieses Prinzip wird gewährleistet, daß der Nutzer jedem MAMO-Feldbefehl, jeder Gruppe von MAMO-Feldbefehlen oder einem ganzen

MAMO-Programm eine "eigene PRAN-Behandlungsroutine zuordnen kann. • Konnte die PRAN-Behandlungsroutine den Programmschaden beheben, dann wird der Übergang (9) zum **Block 9** vorgenommen und dort der Befehlszählerstand des unterbrochenen MAMO-Befehls wieder hergestellt und dieser Befehl angesprochen (11).

Gelingt der PRAN-Behandlungsroutine die gestellte Aufgabe nicht, dann wird mit einem Programmende-SVC-Befehl die Aktivität dem Betriebssystem zurück gegeben (8). In der ABEND-Routine des Steuerprogramms wird der BESETZT-Zustand des MAMO mit dem Befehl MUC aufgehoben. Ist bei Auftreten von CC = 1 keine PRAN-Behandlungsroutine vorgesehen, sorgt die Anschlußroutine für den Übergang 8 zum Betriebssystem. Die Sonderbehandlung der arithmetischen Ausnahme bei MAMO-Feldbefehlen führt zu einer wesentlichen Beschleunigung dieser Prozedur, da der Weg über das Betriebssystem vermieden wird. Im Verarbeitungsprozeß von Datenfeldern kann die PRAN-Behandlung hochfrequent auftreten (z. B. Mantissenverlust, Exponentenunterschreitung). Eine Reparaturmöglichkeit besteht in diesem Fall darin, die entsprechenden Resultatkomponenten durch echte Nullen zu ersetzen.

4.3. Servicebedienung und Diagnose

Die Servicebedienung des MAMO erfolgt über die Bedieneinheit EC 7069 oder den BSP EC7069M. Die einzelnen Funktionseinheiten des MAMO werden über spezielle Servicebedienungsbilder funktionell dargestellt.

Die wichtigsten Register, Zähler, Steuertrigger werden standardmäßig angezeigt. Jedes MAMO-Service-Bild enthält im unteren Bildbereich einen variablen Teil, über den man jeden weiteren MAMO-Trigger über eine bestimmte Anzeigeadresse anzeigen kann. Da der MAMO über eine orthogonale Datenshifteinrichtung, den Diagnosebus, verfügt, können zu Testzwecken im Taktzyklus- bzw. Taktphasenbetrieb alle MAMO-Trigger auf einen willkürlich wählbaren Wert gesetzt werden. Alle diese Manipulationen erfolgen von der Bedieneinheit oder dem BSP aus.

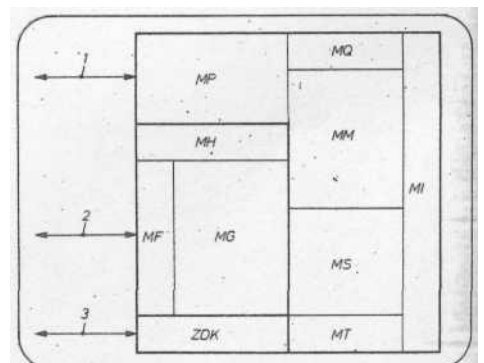


Abb. 19 MAMO-Blockstruktur

Ebenfalls sind in den genannten Taktsteuer-Modi *alle* MAMO-RAM-Speicherplätze anzeig- bzw. beschreibbar. Diese genannten Eigenschaften sind Basis der realisierten Testmittel. Der Test erfolgt schwerpunktmäßig durch

- **Mikrotest 1 (MT1)**

Ein ZVE-Mikrogrammpaket testet von außen das ZVE/MAMO-Interface und einige Grundfunktionen, die im wesentlichen der Kommunikation ZVE/MAMO innerhalb der nachfolgenden Tests dienen.

- **Mikrotest 2 (MT2)**

MAMO-Testmikrogramme testen den MAMO funktionseinheitenweise nach dem Prinzip des aufbauenden Tests. Beide Testgruppen laufen unter Steuerung des Mikrotestmonitors. Aufbauend auf den Basistests MT1 und MT2 arbeiten die Testprogramme. Ihre hauptsächliche Aufgabe ist, die Übereinstimmung der MAMO-Funktionen mit den Vorgaben der MAMO-Operationsprinzipien zu kontrollieren und darüber hinaus die Testschärfe des Basistests zu ergänzen.

5. Voraussetzungen zum Betrieb des MAMO im OS/ES

5.1. Logische Mikroprogrammspeicher LMPS, LTMPs

LMPS:

Der MAMO besitzt ein eigenes Mikroprogrammsteuerwerk. Neben einem ROM-Speicher, der über 32 Mikrobefehle verfügt, sind alle übrigen Mikroprogramme in einem Speicherbereich (LMPS) untergebracht, welcher sich am Ende des HS befindet. Der MAMO nutzt einen eigenen RAM-Mikrogrammpuffer mit einer Kapazität von 512 Mikrobefehlen (je 4 Byte). Die ersten etwa 200 Mikrobefehle bilden den Kernteil, der zum Abarbeiten aller Algorithmen benötigt wird. Dieser Mikroprogrammkernel wird nur einmal innerhalb des MAMO-IPL aus dem LMPS in den Mikroprogrammpuffer geladen. Damit ist der MAMO im BEREIT-Zu-stand. Wird dem MAMO ein definierter Untercode übermittelt, dann lädt er das entsprechende Mikroprogramm aus dem LMPS in den restlichen ladbaren Teil des Puffers. Die ladbaren Teile werden *variable Komplexe* genannt und haben bei einer LMPS-Version immer eine konstante Länge, d. h. 512 Mikrobefehle minus der Länge des Kernmikrogramms. In einem variablen Komplex werden bis zu acht Algorithmen untergebracht. Ein zu einem Untercode gehörendes Mikroprogramm kann in mehreren variablen Komplexen enthalten sein, um bei bestimmten Befehlsfolgen in bestimmten Anwenderprogrammen die Umladezeiten einzusparen. Um dem MAMO mitzutei-

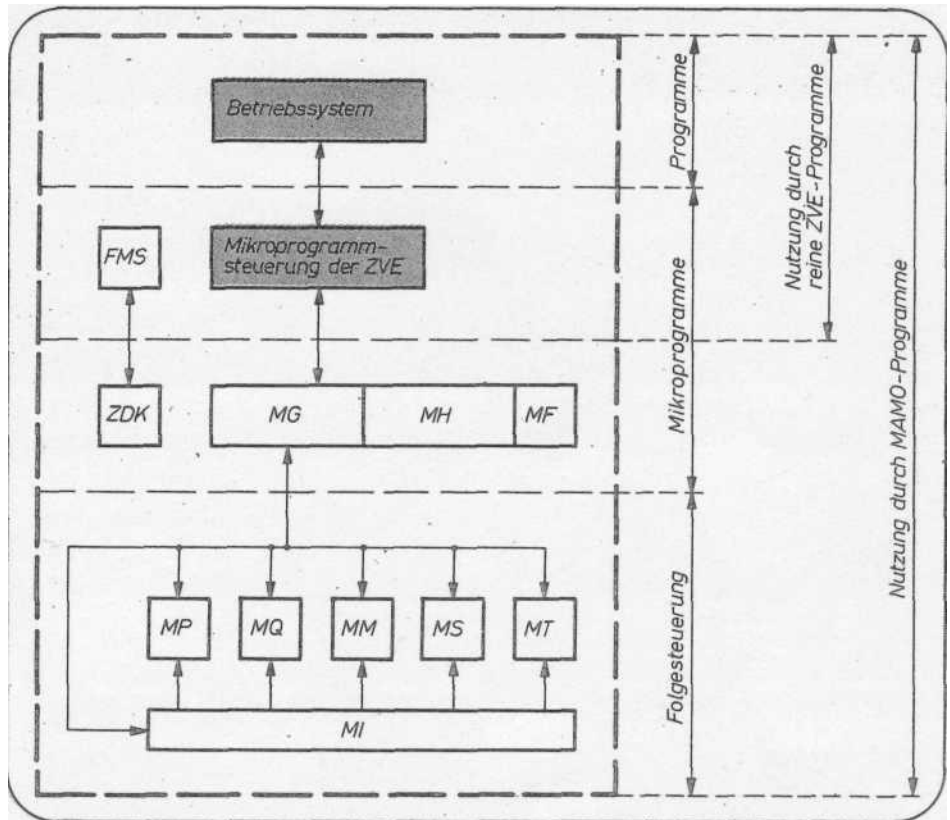


Abb. 20 Steuerebenen

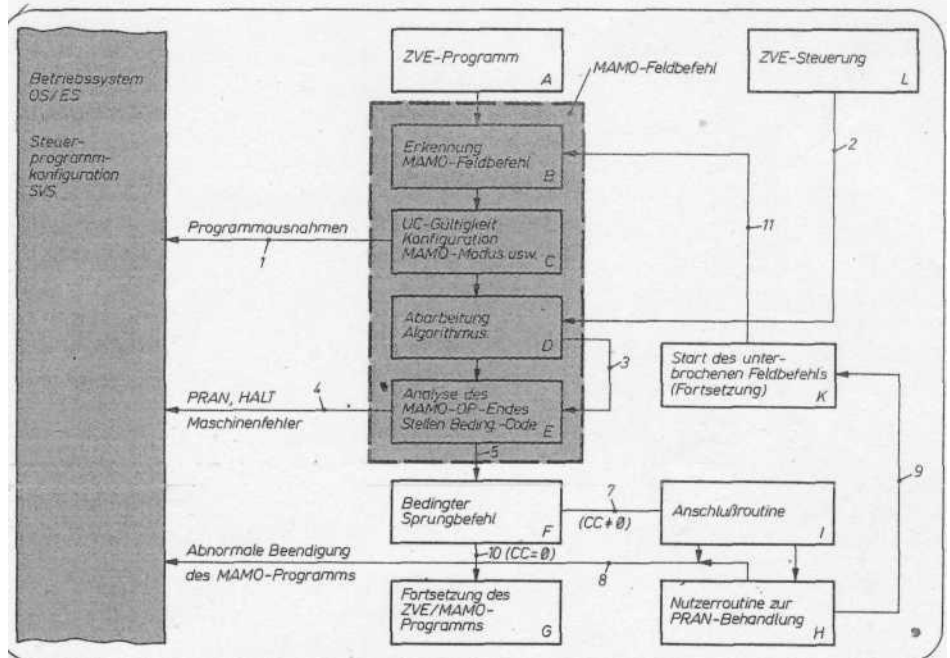


Abb. 21 Wichtigste Aktivitäten beim Ablauf eines MAMO-Feldbefehls

len, welche Untercode durch die im variablen Komplex enthaltenen Mikroprogramme realisiert werden, ist an jeden variablen Komplex noch ein -Untercodeverzeichnis mit einer Länge von zwei Doppelworten gekettet. Dieses Untercodeverzeichnis lädt sich der MAMO in einen kleinen RAM-Speicher. Bei weiteren MAMO-Befehlsstarts prüft der MAMO, ob das Mikroprogramm für den neu zu startenden MAMO-Feldbefehl bereits in dem geladenen variablen Komplex enthalten ist. Ist das nicht der Fall, dann lädt der MAMO den erforderlichen

variablen Komplex selbsttätig aus dem LMPS in seinen Mikroprogrammpuffer. Um die Übertragungsprozesse der LMPS-Datei von einem externen Datenträger in den HS hinreichend sicher zu kontrollieren, werden alle Daten des LMPS doppelwortweise in einer bestimmten Weise addiert. Die entstandene Summe ist komplementiert auf einer bestimmten Adressposition im LMPS eingetragen. Die ZVE besitzt Mikroprogramme, die teils automatisch, teils über den MAMO-

CC	Bedeutung
0	<ul style="list-style-type: none"> - MAMO-Feldbefehlsalgorithmus wurde vollständig regulär abgearbeitet. - Der MAMO ist bereit zur Ausführung eines neuen MAMO-Befehls.
1	<ul style="list-style-type: none"> - Der Algorithmus wurde durch ein Ereignis in seiner Ausführung unterbrochen. - Der MAMO ist besetzt und sein Algorithmus jederzeit fortsetzbar.
2	<ul style="list-style-type: none"> - Die Ausführbarkeit des Algorithmus ist aufgrund bestimmter Ereignisse nicht möglich. - Der MAMO ist nicht besetzt und ist bereit zur Ausführung eines neuen MAMO-Befehls.
3	<ul style="list-style-type: none"> - Verbotener Bedingungscode.

Abb. 22
Darstellung des Bedingungscode beim Operationsende eines MAMO Befehls durch das Mikroprogramm, Block E

Befehl LSM angesprochen werden können. Diese sind in der Lage, eine Testaddition mit dem gleichen Algorithmus auszuführen. Nach einer Testaddition muß als Testsumme über den LMPS-Inhalt ein 64-Bit-Einsvektor entstehen, damit der gesamte LMPS als fehlerfrei akzeptiert wird.

Ein leicht abgewandeltes Verfahren wird auch zum Prüfen der ständigen Mikroprogrammloadprozesse zwischen LMPS und MAMO-Mikroprogramm puffer angewandt. Tritt ein Prüfsummenfehler im MAMO ein, dann wird der MAMO-Ablauf unterbrochen und ein « Meldung an die ZVE gesandt. Das bereits genannte ZVE-Prüfprogramm kontrolliert daraufhin den LMPS im HS. Ist er in Ordnung, dann wird der MAMO als fehlerhaft gemeldet. Ist der LMPS defekt, ohne daß die ECC-Einrichtung angesprochen hat, wird ein mikroprogrammtechnisch erzeugter Maschinenfehler mitgeteilt.

LTMPs (Logischer Test-Mikroprogramm Speicher):

Ebenfalls am Ende des HS, hinter der LMPS-Datei, befindet sich ein Bereich, der zum Testen des MAMO dient. Er enthält:

- ladbare MAMO-Testmikroprogramme
- ladbare ZVE-Mikroprogramme
- Testdaten, Solldaten
- Logout/Login-Bereich
- Steuertabellen zum Auffinden der einzelnen Dateien.

Die Mikrotestphilosophie des MAMO sieht vor, daß die gesamte Testorganisation von einem mikroprogrammierten ZVE-MAMO-Testmonitor ausgeführt wird. Dieser Testmonitor ist ein Bestandteil des ZVE-Mikrotest-Komplexes. Die Aufgaben des Testmonitors sind:

- Start der einzelnen Testschritte im MAMO
- Versorgung der Testschritte mit ihren speziellen Test- und Solldaten
- Soll/Ist-Vergleich nach jedem Testschritt und Eintragung der Testergebnisse im Fehlerfall in den Diagnosepuffer der ZVE.

5.2. MAMO-IPL

Das MAMO-IPL-Programm ist ein systemunabhängiges Programm zum Initialisieren des MAMO. Die *Betriebssystem-*

unterstützung für den MAMO setzt voraus, daß zum Zeitpunkt der Initialisierung des Betriebssystems der MAMO bereits funktionsfähig ist. Das heißt, daß er in der Lage ist, in Verbindung mit einem bereits im HS befindlichen LMPS MAMO-Befehle auszuführen und daß in Verbindung mit dem ebenfalls bereits geladenen LTMPs der MAMO-Mikrotest arbeitsfähig ist.

Diese Arbeitsbereitschaft des MAMO wird durch das Programm MAMO-IPL herbeigeführt. Es arbeitet systemunabhängig und wird vor dem Start der Betriebssysteminitialisierung ausgeführt. Das selbstladende Programm MAMO-IPL erledigt das Laden der Dateien LMPS und LTMPs von einem Magnetband in den oberen Adreßbereich des HS und das Initialisieren des MAMO. Bevor das Mikroprogramm-Kernladen und die MAMO-BEREIT-Emschaltung erfolgen, führt das Programm unter anderem Testsummenprüfungen für den LMPS und LTMPs sowie einen vollständigen Mikrotest in ZVE und MAMO durch. Nach Beendigung des Programms MAMO-IPL hat der MAMO den Zustand „BEREIT“. „GESPERRT“ und „NICHT BESETZT“. Das darauffolgende Betriebssystem-IPL darf nur mit „IPL ohne Löschen des HS“ ausgeführt werden, um die geladenen Dateien LMPS und LTMPs nicht zu zerstören,

5.3. Steuerbefehle

Die im Pkt. 3.1. genannten Algorithmen sind lediglich der Teil der Funktionen, die ausschließlich auf die Problemlösung des Anwenders orientiert sind. Um den MAMO im Betriebssystem wirkungsvoll zu unterstützen, sind folgende sechs privilegierte Steuerbefehle geschaffen worden.

- MUC: (MAMO-USING-CONTROL) Dieser Befehl dient dem Betriebssystem dazu, einem Problemprogramm die Benutzung des MAMO zu gestatten bzw. zu verbieten. Der Befehl läuft auch bei nicht einkonfiguriertem MAMO ab.
- MST: (MAMO-STATUS-TEST) Diesen Befehl nutzt das Betriebssystem, um die Verwendbarkeit des MAMO festzustellen.
- LSM: (LOAD-STATUS-MAMO)

Dieser Befehl hat für den MAMO eine ähnlich weitreichende Bedeutung wie der Diagnosebefehl für den Universalprozessor. Er wird sowohl vom Betriebssystem als auch von den autonomen Testprogrammen genutzt. Die Befehlsfunktion wird durch einen besonderen Parameterblock spezifiziert. Einige Funktionen seien genannt:

- „KERN LADEN“ (IPL-Programm)
- LMPS/LTMPs-Summenprüfung
- Login/Logout
- Datentransfer zwischen LMPS/LTMPs-Bereich und dem vom Betriebssystem genutzten HS-Teil
- Aktivierung beliebiger MAMO-Mikroprogramme usw.

- SMC: (SET-MAMO-COUNTER) Mit diesem Befehl lädt das Betriebssystem den MAMO-Zähler, der in einem MAMO-Programm die Anzahl der MAMO-Befehlsstarts zählt. Der MAMO-Zähler zählt auch weiter, wenn ein unterbrochener MAMO-Befehl fortgesetzt wird.

- SMT: (SET-MAMO-TIMER) Dieser Befehl gestattet dem Betriebssystem das Voreinstellen der MAMO-Uhr.

- MMC: (MAMO-MONITOR-CALL) Mit diesem Befehl erreicht das Betriebssystem, daß sich der MAMO nach vollständiger Abarbeitung des laufenden Algorithmus mit einer MONITOR-CALL-Unterbrechung beim Betriebssystem zu rückmeldet, damit ein MAMO-Programm höherer Priorität gestartet werden kann.

Zusätzlich zu diesen sechs privilegierten Befehlen sind noch drei nichtprivilegierte Steuerbefehle verwirklicht:

- STMC: (STORE-MAMO-COUNTER) Mit diesem Befehl kann auch der Nutzer die Anzahl der in seinem Programm bis zu diesem Zeitpunkt ausgeführten MAMO-Befehlsstarts abfragen, um eine Information über den Grad des Einflusses der Unterbrechungen auf die MAMO-Leistung zu erhalten.
- STMT: (STORE-MAMO-TIMER)

über diesen Befehl mißt das Problemprogramm den reinen MAMO-Zeitanteil in seinem Programm. Für das Betriebssystem dient er zu Abrechnungszwecken. Außerdem kann das autorisierte Problemprogramm feststellen, ob der MAMO besetzt ist

- RBB: (RESET-BUSY-BIT) Mit diesem Befehl kann ein autorisiertes Problemprogramm im Falle der Unmöglichkeit, nach einer fortsetzbaren Programmausnahme den MAMO-Befehl tatsächlich fortzusetzen, den BESETZT-Zustand aufheben.

6. Betriebssystemunterstützung ■

6.1. Unterstützung der Programmierung von MAMO-Problemen

Die Programmierung des MAMO erfolgt zur Zeit in Assemblersprache. Mit Hilfe

von speziellen Assembler-Makros wird die Arbeit mit dem Mitteilungsblock MIB erleichtert. Die erforderlichen Makrodefinitionen sind in der Systembibliothek SYS1. MACLIB enthalten. Die Makroanweisungen für die MAMO-Befehle werden bei der Programmierung wie ZE-Befehle von SS-Format unter Benutzung der mnemonischen Befehls-codes codiert. Die Operandenadressen weisen auf den MIB und den PAB. Mit dem Makro MPB wird der Parameterblock PAP und mit dem Makro MIB der Mitteilungsblock MIB generiert. Um einen bereits bestehenden Parameterblock für nachfolgende MAMO-Feldbefehle weiter-nutzen zu können, ermöglicht der Makro MPB eine Modifikation seines Inhalts. Das analoge Vorgehen wird mit dem Makro MIB für den Mitteilungsblock MIB ermöglicht, um den MAMO auch in Programmen verwenden zu können, die in den höheren Programmiersprachen FORTRAN, COBOL oder PL/1 geschrieben sind, müssen die jeweiligen Anschlußmakros verwendet werden, um die MAMO-Assembler-routinen in den jeweiligen Programmverband zu integrieren.

6.2. Funktionen im Steuerprogramm

Die Unterstützung des MAMO in der EC-1055/EC-1055M-Konfiguration erfolgt generierbar in der Steuerprogramm-konfiguration SVS des OS/ES. Die Steuerprogrammkonfiguration SVS gestattet einen hinreichend sicheren Schutz gegen beabsichtigte oder unbeabsichtigte Beschädigung der für den MAMO-Prozess im System lebenswichtigen Steuerdateien LMPS und LTMPs am Bereichs-ende des HS durch Problemprogramme. Diese Eigenschaft wird dadurch erreicht, daß die im Systeminitialisierungsprozess erzeugte Seitenrahmentabelle die Seiten des Steuerdatenbereiches am HS-Ende nicht enthält.

Da durch verdeckte Maschinenfehler diese Schutzmaßnahmen trotzdem unwirksam sein können, werden bei den dynamischen Mikroprogrammladeprozessen ständig die Summenprüfungen bei jedem Übertragungsprozess ausgeführt und damit ein solcher fehlerhafter Einfluß erkannt.

Die MAMO-Feldbefehle werden seitens des Betriebssystems bezüglich ihrer Anwendbarkeit den ZE-Befehlen gleichgestellt. Das Steuerprogramm vermittelt den MAMO an die im System laufenden Aufgaben nach ähnlichen Kriterien, wie sie für die Aufgabenauswahl im allgemeinen zutreffen. Falls der MAMO nicht besetzt ist, wird er bei Aufruf eines Feldbefehls der aktiven Aufgabe zugeordnet. Die Zuordnung erfolgt mindestens für die Dauer eines MAMO-Feldbefehls, da der MAMO erst nach Beendigung des Be-

fehls mit CC = 0,2 nicht besetzt hinterlassen wird. Zum CC = 1 gilt das im Pkt. 4.2. Gesagte.

Multiprogrammbetrieb von MAMO-Aufgaben

Der Multiprogrammbetrieb des MAMO wird mit einem eigens geschaffenen Hilfsmittel, dem GESPERRT-Modus, verwirklicht.

Um MAMO-Programme entsprechend ihrer Priorität berücksichtigen zu können, ist es zunächst wichtig, das Betriebssystem von ihrem Erscheinen im System zu informieren, damit die neue MAMO-Aufgabe entsprechend ihrer Priorität in die Warteschlange eingeordnet werden kann.

Da sich MAMO-Aufgaben äußerlich von reinen ZVE-Aufgaben nicht unterscheiden, muß ein Mittel gefunden werden, daß nur vom Betriebssystem autorisierten Aufgaben eine MAMO-Benutzung ermöglicht. Dieses Mittel stellt der GESPERRT-Modus dar. Die Funktion ist folgende:

- Nach Einschalten der EC 2655 oder EC2655M ist der MAMO-GESPERRT-Modus eingeschaltet. Jeder Versuch eines Programms, den MAMO mit einem nicht-privilegierten MAMO-Befehl anzusprechen, führt zu einer Operationsausnahme.

- Ist die MAMO-Betriebssystemunterstützung generiert, wird der Operationscode des die Programmausnahme auslösenden Befehls analysiert. Ist der Operationscode 'EI' und der MAMO im BE-REIT-Zustand (siehe MAMO-IPL), wird mit dem MAMO-Steuerbefehl MUC der GESPERRT-Modus ausgeschaltet und, wenn der MAMO im NICHT-BESETZT-Zustand ist, die entsprechende Aufgabe sofort, wieder gestartet. Das laufende PSW hat nun für die Laufzeit des Programms freien Zugang zum MAMO. Wird diese Aufgabe unterbrochen, läßt sich der MAMO durch das Steuerprogramm über MUC sofort wieder in den GESPERRT-Modus versetzen.

- Jedes MAMO-Programm beginnt zu nächst bei seinem ersten Versuch, den MAMO zu benutzen, mit einer Operationsausnahme. Dadurch wird das MAMO-Programm als solches für das Steuerprogramm erkennbar.

Um ein den MAMO besetzendes MAMO-Programm zu veranlassen, nach Beendigung des laufenden MAMO-Feldbefehls die Steuerung zugunsten eines MAMO-Programms höherer Priorität abzugeben, wird vom Steuerprogramm mit dem MAMO-Steuerbefehl MMC ein MAMO-RUF-Trigger eingeschaltet. Beendet ein MAMO-Feldbefehl seine Operation, ohne den MAMO besetzt zu hinterlassen, dann löst er bei eingeschaltetem MAMO-RUF-Trigger eine MAMO-Call-Unterbrechung aus, die zu einer umgehenden Aktivierung des MAMO-Programms mit Priorität führt.