

PDV-E 134
Oktober 1979

PDV-Entwicklungsnotizen

Anpassung von CIMIC / P
an BASIS-PEARL

B. F. Eichenauer, R. Henn, K. Lucas, A. Zeh
GPP, Gesellschaft für Prozeßrechnerprogrammierung mbH,
München

Kernforschungszentrum Karlsruhe

PDV-Berichte

Die Kernforschungszentrum Karlsruhe GmbH koordiniert und betreut im Auftrag des Bundesministers für Forschung und Technologie das im Rahmen der Datenverarbeitungsprogramme der Bundesregierung geförderte Projekt Prozeßlenkung mit Datenverarbeitungsanlagen (PDV). Hierbei arbeitet sie eng mit Unternehmen der gewerblichen Wirtschaft und Einrichtungen der öffentlichen Hand zusammen. Als Projektträger gibt sie die Schriftenreihe PDV-Berichte heraus. Darin werden Entwicklungsunterlagen zur Verfügung gestellt, die einer raschen und breiteren Anwendung der Datenverarbeitung in der Prozeßlenkung dienen sollen.

Der vorliegende Bericht dokumentiert Kenntnisse und Ergebnisse, die im Projekt PDV gewonnen wurden.

Verantwortlich für den Inhalt sind die Autoren. Die Kernforschungszentrum Karlsruhe GmbH übernimmt keine Gewähr insbesondere für die Richtigkeit, Genauigkeit und Vollständigkeit der Angaben, sowie die Beachtung privater Rechte Dritter.

Druck und Verbreitung:

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640 7500 Karlsruhe 1

Bundesrepublik Deutschland

PDV-E 134

PROJEKT PROZESSLENKUNG MIT DV-ANLAGEN
ENTWICKLUNGSNOTIZ PDV-E 134

ANPASSUNG VON CIMIC/P
AN BASIS-PEARL

VON

B.F. EICHENAUER, R. HENN,

K. LUCAS, A. ZEH

GPP, GESELLSCHAFT FÜR PROZESSRECHNER-
PROGRAMMIERUNG MBH, MÜNCHEN

OKTOBER 1979

Zusammenfassung:

Der vorliegende Bericht beschreibt die Änderungen von CIMIC/P gegenüber der Beschreibung GPP/5/77. Diese Änderungen wurden bei der Anpassung des GPP-PEARL-Compilers an BASIS-PEARL gemäß DIN-Entwurf 66253 erforderlich.

Inhalt

gestrichen:

- 4.2.7 Vereinbarung von Files
- 4.2.8 Vereinbarung von Devices

neu aufgenommen:

- 4.2.13 Vereinbarung von Datenstationen
- 4.2.14 Vereinbarungen zum Testen und Bedienen

ändern:

- 9.1 Verwaltung der Datenwege
- 9.2 Öffnen und Schließen von DATIONS
- 9.3 Anweisungen zur Übertragung
- 9.4 Format-Listen
- 9.5 Daten-Listen

entfällt:

- 9.6 Prozess-Ein-/Ausgabe

3.3.1 Elementare Datentypen

Änderung

Ein Operand vom Typ ADDR enthält eine Speicheradresse im
ADDR Daten- oder Anweisungsbereich eines Programms bzw. die
Zugriffsfunktion zu einem Objekt.

3.4.1 Konstanten

Zusatz bei REAL-CONSTANT

Die Notation geschieht zur Basis 10. Der Dezimalpunkt wird links vor der ersten Ziffer angenommen (normierte Darstellung).

Änderung bei LABEL-CONSTANT

LABEL-CONSTANT:

'R' LABEL [' : BLOCK-LEVEL].

LABEL: SYMBOL.

BLOCK-LEVEL: INTEGER.

Durch BLOCK-LEVEL wird die Blocktiefe der entsprechenden Vereinbarung angegeben (siehe 4.2.6). Diese Option wird nur bei Marken verwendet, deren Ansteuerung möglicherweise mit Blockwechsel verbunden ist.

3.4.2 Variablen

Änderung bei REFERENCE

MODIFICATION:

OFFSET + BIT-SELECTOR.

OFFSET:

'(' [FIX-OFFSET] ')'
[SINGLE-SIMPLE-MODE /
STRUCTURE-MODE-IDENTIFICATION /
IRPT-MODE / SIGNAL-MODE /
SEMA-MODE / DATION-MODE].

BIT-SELECTOR:

' BIT ' INTEGER.

FIX-OFFSET:

TERM // ('+' / '-').

TERM:

[INTEGER '**']
SINGLE-SIMPLE-MODE.

4.1 Moduln

Anderung bei Modulaufbau

MODULE:

```
[ 'LINE ' INTEGER EOL ]  
'MODULE ' [ IDENTIFIER ] EOL  
( MODULE-ELEMENT EOL ) $  
'MODEND' EOL.
```

Mit LINE wird der Bezug zum Listing des Quellprogramms hergestellt.

4.2.2 Vereinbarungen globaler Größen

Änderung bei der Zusammenstellung

GLOBAL-DECLARATION:

'DSECT ' EOL

('SPACE ' (

 SIMPLE-DECLARATION /

 SIMPLE-ARRAY-DECLARATION /

 STRUCTURE-DECLARATION /

 STRUCTURE-ARRAY-DECLARATION /

 SEMA-DECLARATION /

 IRPT-DECLARATION /

 IRPT-ARRAY-DECLARATION /

 SIGNAL-DECLARATION /

 SIGNAL-ARRAY-DECLARATION /

 DATION-DECLARATION /

 DATION-ARRAY-DECLARATION)

)•

'ENDBLK DSECT '.

Verbesserung von

SIMPLE-DECLARATION:

```
SINGLE-SIMPLE-MODE ' '  
( GLOBAL-CATEGORY ' ' XSYMB /  
  S-D-CATEGORY ' ' SYMBOL )  
[ ' ' DATA-CONSTANT ].
```

GLOBAL-CATEGORY:

```
[ ' INV, ' ]  
( ' DSECT' / ' RSECT' ).
```

Vereinbarung von Feldern

SIMPLE-ARRAY-DECLARATION:

```
DOPE-MODE ' '  
(GLOBAL-CATEGORY ' ' XSYMB/  
S-D-CATEGORY ' ' SYMBOL)  
DIMENSION-DECLARATION ' '  
'R ' SYMBOL EOL  
'SPACE ' SINGLE-SIMPLE-MODE ' '  
(GLOBAL-CATEGORY/S-D-CATEGORY)  
' ' SYMBOL  
DIMENSION-DECLARATION  
['+' EOL ARRAY VALUE].
```

wobei

DIMENSION-DECLARATION:

```
'(['[INTEGER','] INTEGER','] INTEGER')'.
```

explizit die oberen Grenzen angibt.

Zusatz bei Initialisierung von Feldern

ARRAY-ELEMENT-INITIALIZATION:

(S-MODE / SEMA-MODE) ' '
(GLOBAL-CATEGORY / S-D-CATEGORY) ' '
'(' INTEGER ')'
DATA-CONSTANT.

wobei

S-MODE:

SINGLE-SIMPLE-MODE /
'ADDR'.

Vereinbarung von Feldern von Strukturen

STRUCTURE-ARRAY-DECLARATION:

```
DOPE-MODE ' '  
  (GLOBAL-CATEGORY ' ' XSYMB/  
  S-D-CATEGORY ' ' SYMBOL)  
  DIMENSION-DECLARATION ' '  
  'R ' SYMBOL EOL  
'SPACE ' STRUCTURE-MODE-IDENTIFICATION ' '  
  (GLOBAL-CATEGORY / S-D-CATEGORY)  
  ' ' SYMBOL  
  DIMENSION-DECLARATION  
  ('+' EOL STRUCTURE-ARRAY-VALUE/  
  EOL STRUCTURE-BODY-DECLARATION).
```

Änderung beim Verweis auf weitere Abschnitte
mit Vereinbarungen globaler Größen

INTERRUPT	(Abschnitt 4.2.9)
SIGNAL	(Abschnitt 4.2.10)
SEMA	(Abschnitt 4.2.11)
DATION	(Abschnitt 4.2.13)

4.2.3 Block und Variablenvereinbarung

Änderung bei Vereinbarungen statischer Variablen

STATIC-DECLARATION:

LOCAL-SYMBOL-IDENTIFICATION EOL

```
'SPACE '(  
    SIMPLE-DECLARATION /  
    SIMPLE-ARRAY-DECLARATION /  
    STRUCTURE-DECLARATION /  
    STRUCTURE-ARRAY-DECLARATION /  
    SEMA-DECLARATION /  
    DATION-DECLARATION).
```

Änderung bei LOCAL-BLOCK-DECLARATION

LOCAL-BLOCK-DECLARATION:

```
'BLOCK ' BLOCK-LEVEL ' DISPØ ' EOL  
      (CODE-ELEMENT EOL)*  
'BLEND ' BLOCK-LEVEL ' DISPØ '.
```

wobei

BLOCK-LEVEL: INTEGER.

Um die Schachtelung lokaler Blöcke zu ermöglichen, wird durch die Ganzzahl-Konstante BLOCK-LEVEL die Blocktiefe bezüglich einer Prozedur oder Task festgelegt.

4.2.5 Prozedurvereinbarung

Änderung

PROCEDURE-HEAD:

```
[ 'N'/'Q' ] 'BEGIN '  
    PROCEDURE-HEADING EOL  
( [ 'N'/'Q' ] 'PARAM '  
    (VALUE-P-SPECIFICATION /  
    ADDR-P-SPECIFICATION /  
    ARRAY-P-SPECIFICATION )  
)$  
[ 'N'/'Q'/'R' ] 'PAREND '  
    PROCEDURE-HEADING EOL
```

PROCEDURE-HEADING:

```
[ RESULT-MODE ] ' '  
( G-CATEGORY ' ' XSYMB /  
[ 'R' ] D-CATEGORY ' ' SYMBOL ).
```

VALUE-P-SPECIFICATION:
SINGLE-SIMPLE-MODE ' '
['INV,'] P-CATEGORY.

ADDR-P-SPECIFICATION:
'ADDR' ' ' P-CATEGORY ' '
'REF' ['INV,'] SINGLE-SIMPLE-MODE /
(['INV,'] STRUCTURE-MODE-IDENTIFICATION EOL
STRUCTURE-MODE DISPLAY) /
(DATION-MODE EOL
DATION-MODE-DISPLAY).

wobei

P-CATEGORY:
G-CATEGORY ' ' /
D-CATEGORY ' ' SYMBOL.

Bei der Parameterübertragung wird gemäß dem IDENTICAL-Mechanismus die Zugriffsfunktion übertragen. Durch REF wird beschrieben, welche Typen dabei referiert werden.

ARRAY-P-SPECIFICATION:

```
DOPE-MODE ' ' P-CATEGORY
  ( DIMENSION-SPECIFICATION ' '
    ['INV,'] SINGLE-SIMPLE-MODE) /
  (DIMENSION-SPECIFICATION ' '
    ['INV,'] STRUCTURE-MODE-IDENTIFICATION EOL
    STRUCTURE-MODE DISPLAY) /
  (SINGLE-DIMENSION-SPECIFICATION ' '
    DATION-MODE EOL
    DATION-MODE-DISPLAY).
```

wobef

DIMENSION-SPECIFICATION:

```
'(' [ '[' ' ' , ' ] ' ' ' , ' ] ' ' ' ' )'.
```

SINGLE-DIMENSION-SPECIFICATION:

```
'(' ' ' ' )'.
```

4.2.6 Markenvereinbarungen

Eine Marke wird mit der Anweisung

```
LABEL-DECLARATION:  
[LOCAL-SYMBOL-IDENTIFICATION EOL]  
'LOC ' LABEL [' ' BLOCK-LEVEL].
```

wobei

```
LABEL: SYMBOL.  
BLOCK-LEVEL: INTEGER.
```

eingeführt. Darin stellt SYMBOL die abstrakte Adresse der nächsten Anweisung dar. Mit BLOCK-LEVEL wird die Blocktiefe dieser Vereinbarung angegeben. Diese Option wird nur bei Marken verwendet, deren Ansteuerung möglicherweise mit Blockwechsel verbunden ist.

4.1.7 Vereinbarung von Files

4.1.8 Vereinbarung von Devices

entfällt zu Gunsten 4.2.13

4.2.9 Vereinbarung von Interrupts

Die Deklaration findet nur auf Modul-Ebene statt. Sie ist dabei immer mit einer entsprechenden System-einrichtung verbunden. Der Bekanntheitsgrad ist nicht auf den Modul beschränkt.

IRPT-DECLARATION:

```
IRPT-MODE ' '  
GLOBAL-CATEGORY ' ' XSYMB EOL  
CONNECTION  
'IRPTEND '.
```

IRPT-ARRAY-DECLARATION:

```
DOPE-MODE ' '  
GLOBAL-CATEGORY ' ' XSYMB  
SINGLE-DIMENSION-DECLARATION  
'R ' SYMBOL EOL  
'SPACE ' IRPT-MODE ' '  
GLOBAL-CATEGORY ' ' SYMBOL  
SINGLE-DIMENSION-DECLARATION EOL  
CONNECTION  
'IRPTEND '.
```

Die Spezifikation muß immer einer Deklaration aus einem anderen Modul zugeordnet werden. Es handelt sich dabei um Interrupts, die entweder dem System bekannt sind, oder um dafür frei gewählte Bezeichner.

IRPT-SPECIFICATION:

```
IRPT-MODE ' '  
GLOBAL-CATEGORY ' ' XSYMB EOL  
'IRPTEND '.
```

IRPT-ARRAY-SPECIFICATION:

```
DOPE-MODE ' '  
GLOBAL-CATEGORY ' ' XSYMB  
SINGLE-DIMENSION-SPECIFICATION  
' ' IRPT-MODE EOL  
'IRPTEND '.
```

Andere Formen von Interrupt-Vereinbarungen treten nicht auf:

- Modul-, Task-, Block-local
- formaler Parameter
- Interrupt-channel bei Dations

4.2.10 Vereinbarungen von Signals

Die Deklaration findet nur auf Modul-Ebene statt. Sie ist dabei immer mit einer entsprechenden System-einrichtung verbunden. Der Bekanntheitsgrad ist nicht auf den Modul beschränkt.

SIGNAL-DECLARATION:

```
SIGNAL-MODE ' '  
GLOBAL-CATEGORY ' ' XSYMB EOL  
CONNECTION  
'SIGNEND '.
```

SIGNAL-ARRAY-DECLARATION:

```
DOPE-MODE ' '  
GOBAL-CATEGORY ' ' XSYMB  
SINGLE-DIMENSION-DECLARATION  
'R ' SYMBOL EOL  
'SPACE ' SIGNAL-MODE ' '  
GLOBAL-CATEGORY ' ' SYMBOL  
SINGLE-DIMENSION-DECLARATION EOL  
CONNECTION  
'SIGNEND '.
```

Die Spezifikation muß immer einer Deklaration aus einem anderen Modul zugeordnet werden. Es handelt sich dabei um Signals, die entweder dem System bekannt sind, oder um dafür frei gewählte Bezeichner.

SIGNAL-SPECIFICATION:

```
SIGNAL-MODE ' '  
GLOBAL-CATEGORY ' ' XSYMB EOL  
'SIGNEND '.
```

SIGNAL-ARRAY-SPECIFICATION:

```
DOPE-MODE ' '  
GLOBAL-CATEGORY ' ' XSYMB  
SINGLE-DIMENSION-SPECIFICATION  
' ' SIGNAL-MODE EOL  
'SIGNEND '.
```

Andere Formen von Signal-Vereinbarungen treten nicht auf:

- Modul-, Task-, Block-local
- formaler Parameter
- Signal-channel bei Dations
- Standard-Signals

4.2.11 Vereinbarungen von Semaphoren

Die Deklaration findet nur auf Modul-Ebene statt. Dabei ist eine Initialisierung mit Werten vom Typ INT möglich.

```
SEMA-DECLARATION:  
  SEMA-MODE ' '  
  (GLOBAL-CATEGORY ' ' XSYMB /  
  S-CATEGORY ' ' SYMBOL)  
  [ ' ' DATA-CONSTANT ].
```

Felder von Semaphoren sind nicht vorgesehen.

Die Spezifikation muß immer eine Deklaration aus einem anderen Modul zugeordnet werden.

SEMA-SPECIFICATION:
SEMA-MODE ' '
GLOBAL-CATEGROY ' ' XSYMB.

Andere Formen von Sema-Vereinbarungen treten nicht auf:

- formaler Parameter.

4.2.12 Formatvereinbarungen

Änderung

In BASIS-PEARL können auf Modul-Ebene Listen von Formaten (controls) definiert werden. Diese werden im Körper von Tasks und Prozeduren von Ein/Ausgabe-Anweisungen angesprochen (sog. R-Format). Ihre Category ist demnach STATIC.

Werden die Formate (controls) im Zusammenhang mit einer Ein-Ausgabe-Anweisung definiert, ist ihre Speicherklasse CODE.

LIST-CATEGORY:

('STATIC' / 'CODE').

Der generelle Aufbau ist in Abschnitt 9.4 beschrieben.

4.2.13 Vereinbarung von Datenstationen

Die Deklaration findet nur auf Modul-Ebene statt.

DATION-DECLARATION:

```
DATION-MODE ' '  
(GLOBAL-CATEGORY ' ' XSYMB /  
S-CATEGORY ' ' SYMBOL ) EOL  
DATION-MODE-DISPLAY.
```

DATION-ARRAY-DECLARATION:

```
DOPE-MODE ' '  
GLOBAL-CATEGORY ' ' XSYMB  
SINGLE-DIMENSION-DECLARATION  
'R ' SYMBOL EOL  
'SPACE ' DATION-MODE ' '  
GLOBAL-CATEGORY ' ' SYMBOL  
SINGLE-DIMENSION-DECLARATION EOL  
DATION-MODE-DISPLAY.
```

Felder von DATION

- a) sind nur eindimensional
- b) nur in Verbindung mit GLOBAL oder DISPLAY.

Die Spezifikation muß immer einer Deklaration aus einem anderen Modul zugeordnet werden.

DATION-SPECIFICATION:

```
DATION-MODE ' '  
GLOBAL-CATEGORY ' ' XSYMB EOL  
DATION-MODE-DISPLAY.
```

DATION-ARRAY-SPECIFICATION:

```
DOPE-MODE ' '  
GLOBAL-CATEGORY ' ' XSYMB  
SINGLE-DIMENSION-SPECIFICATION  
' ' DATION-MODE EOL  
DATION-MODE-DISPLAY.
```

Weiterhin können DATION auch als formale Parameter definiert werden.

Blocklocale Vereinbarungen sind nicht vorgesehen.

Die Beschreibung der DATION-Attribute und der Channel-Topologie sowie die Zuordnung zu anderen DATION ist zusammengestellt in:

DATION-MODE-DISPLAY:

```
'DTYPE ' DATION-TYPE ' '  
      (GLOBAL-CATEGORY / S-D-CATEGORY)  
      ' ' EOL  
'DCONT' [DATION-TRANSFER]  
      ' ' TRANSFER-TYPE EOL  
[ATTACHED]  
'DATEND '.
```

Charakteristika des Aufbaus sowie Einschränkungen des Zugriffs sind beschrieben in:

DATION-TYPE:

```
('INP' / 'OUT' / 'IOP' )  
['.' ( 'DIR' / 'FOR' / 'BACK' )  
  '.' ( 'CYC' / 'NOCYC' )  
  '.' ( 'STRM' / 'NOSTRM' )]  
['.' 'CTR.ALL' ] .
```

Die Übertragungseinheit und der Bezug zum Aufbau des Datenbereiches wird beschrieben durch

```
DATION-TRANSFER:  
    DATION-DIMENSION  
    ['.' TRF-UNIT].
```

wobei

```
DATION-DIMENSION:  
    '(' [INTEGER / ''']  
    ['.' [INTEGER / ''']  
    ['.' [INTEGER / ''']]')'.
```

```
TRF-UNIT:  
    'TFU' ['.MAX'] .
```

Der Datentyp der Übertragungseinheiten wird definiert durch:

TRANSFER-TYPE:
EXTERNAL-TYPE /
INTERNAL-TYPE.

wobei

EXTERNAL-TYPE:
('ALPHIC' / 'BASIC') ' ' /
(GLOBAL-CATEGORY / S-D-CATEGORY) ' ' .

INTERNAL-TYPE:
SINGLE-SIMPLE-MODE ' ' /
(GLOBAL-CATEGORY / S-D-CATEGORY) ' ' /
STRUCTURE-MODE-IDENTIFICATION ' ' /
(GLOBAL-CATEGORY / S-D-CATEGORY) ' ' EOL
STRUCTURE-MODE-DISPLAY.

Zuordnungen zu anderen DATION werden beschrieben in:

ATTACHED:
CREATED /
CONNECTION.

Wenn eine DATION auf einer anderen eingerichtet ist:

CREATED:
'UPON ' DATION-MODE ' '
G-CATEGORY ' ' XSYMB
['(' INTEGER ')'].

Wenn DATION untereinander zusammenschaltet sind, werden die Anschlußpunkte beschrieben:

CONNECTION:
('CONN ' CONNECTION-POINT
' ' ('FROM' / 'TO' / 'FRTO' / 'EQUAL')
' ' CONNECTION-POINT EOL)\$.

Der Anschlußpunkt wird beschrieben:

CONNECTION-POINT:

XSYMB ['/' INTEGER]
[',' INTEGER]

/ XSYMB '(' INTEGER ')'

Darin bedeuten die Ganzzahlen von links nach rechts:

- Geräteindex
- Nummer des Anschlusses

4.2.14 Vereinbarungen zum Testen und Bedienen

- a) Zuordnung von symbolischem Bezeichner zu
Quellidentifizier

LOCAL-SYMBOL-IDENTIFICATION:

```
'IDENT ' SYMBOL ' '  
      'EQUAL' ' ' [XSYMB] .
```

Diese Pseudoanweisung steht vor jeder Definition
eines symbolischen Bezeichners.

- b) Markieren von Ablaufblöcken

EXEC-BLOCK-DECLARATION:

```
'ABLOCK ' BLOCK-NUMBER ' ' D-CATEGORY ' ' EOL  
(LABEL-DECLARATION/  
  STATEMENT/  
  LOCAL-BLOCK-DECLARATION)$  
'ABLEND ' BLOCK-NUMBER ' ' D-CATEGORY ' ' .
```

Diese werden durch

BLOCK-NUMBER: INTEGER .

fortlaufend numeriert.

.3 Spezifikation globaler Größen

Neuer Text

GLOBAL-SPECIFICATION:

'REFER ' EOL

('SPCFY '(

SIMPLE-SPECIFICATION /
SIMPLE-ARRAY-SPECIFICATION /
STRUCTURE-SPECIFICATION /
STRUCTURE-ARRAY-SPECIFICATION /
SEMA-SPECIFICATION /
IRPT-SPECIFICATION /
IRPT-ARRAY-SPECIFICATION /
SIGNAL-SPECIFICATION /
SIGNAL-ARRAY-SPECIFICATION /
DATION-SPECIFICATION /
DATION-ARRAY-SPECIFICATION /
TASK-SPECIFICATION /
PROCEDURE-SPECIFICATION)

)

'ENDBLK REFER '.

Spezifikation von Strukturen

STRUCTURE-SPECIFICATION:

```
STRUCTURE-MODE-IDENTIFICATION ' '  
GLOBAL-CATEGORY ' ' XSYMB EOL  
STRUCTURE-BODY-SPECIFICATION.
```

STRUCTURE-ARRAY-SPECIFICATION:

```
DOPE-MODE ' '  
GLOBAL-CATEGORY ' ' XSYMB  
DIMENSION-SPECIFICATION  
' ' STRUCTURE-MODE-IDENTIFICATION EOL  
STRUCTURE-BODY-SPECIFICATION.
```

wobei

STRUCTURE-BODY-SPECIFICATION:

```
('STCELM ' SINGLE-SIMPLE-MODE ' '  
GLOBAL-CATEGORY ' '[XSYMB]EOL )$  
'STCEED ' SINGLE-SIMPLE-MODE ' '  
GLOBAL-CATEGORY ' '[XSYMB].
```

D

6.1 Monadische Operatoren

Zusatz

'ABS ' SINGLE-ARITHMETIC-MODE

Der Absolutwert des Operanden wird ermittelt und in den Operandenstack gebracht.

'ROUND ' REAL-INT-MODE

Der Operand vom Typ "rationale Zahl" wird zur nächsten ganzen Zahl aufgerundet und das Ergebnis im Operandenstack gespeichert.

'FIT ' (ARITHMETIC-MODE / STRING-MODE)

Der Operand wird (in Genauigkeit bzw. Länge) in den rechts vom Komma stehenden Mode gewandelt und das Ergebnis im Operandenstack abgelegt.

'TRUNC ' REAL-INT-MODE

Der Operand vom Typ "rationale Zahl"
wird zur nächsten ganzen Zahl abgerundet
und das Ergebnis im Operandenstack
gespeichert.

6.2 Dyadische Operatoren

Änderung

('BAND' / 'BOR') ' ' (SINGLE-INT-MODE / BIT-MODE)
'BNEQV' BIT-MODE

noch

BIT-MODE : SINGLE-BIT-MODE ', ' SINGLE-BIT-MODE

'SHIFTL' ['(R)'] ' '
(SINGLE-INT-MODE /
STRING-AND-INT-MODE)

'SHIFTR' ['(R)'] ' '
(SINGLE-INT-MODE /
STRING-AND-INT-MODE)

FIT ist zu streichen.

1.1 Sprunganweisungen

Änderung

```
'JMP ' [CONDITION-CODE]  
' ' (LABEL-CONSTANT/VARIABLE).
```

7.3 ON - Statement

Anderung

```
'ONBEG ' EOL
      ( (CODE-ELEMENT EOL )$
        'ON ' SIGNAL-MODE ' ' G-CATEGORY ' '
        XSYMB ['(') ' SIGNAL-MODE ] EOL ) *
      ( CODE-ELEMENT EOL )$
'ONEND '.
```

STANDARD-SIGNAL entfällt.

7.4 Prozeduraufruf

Anderung

Jeder aktuelle Parameter wird durch

```
[ 'N' / 'Q' ] 'ARGIS '  
  ( (SINGLE-SIMPLE-MODE  
    'ADDR' / DATION-MODE /  
    STRUCTURE-MODE-IDENTIFICATION) ' '  
    [ OPERAND ] ) /  
    (DOPE-MODE ' '  
    ARRAY-IDENTIFICATION).
```

spezifiziert.

Bei der Übergabe von Feldern tritt an Stelle des einfachen OPERANDen eine

```
ARRAY-IDENTIFICATION:  
  G-S-D-CATEGORY ' ' (XSymb/SYMBOL)  
  (DIMENSION-SPECIFICATION ' '  
  (SINGLE-SIMPLE-MODE /  
  STRUCTURE-MODE-IDENTIFICATION )) /  
  (SINGLE-DIMENSION-SPECIFICATION  
  ' ' DATION-MODE).
```

Beispiel für Prozeduraufruf

Anderung von a)

CIMIC/P-Äquivalent:

```
NCALL ...  
LOAD INT.6 DISPØ $L70;  
FIT INT.6,INT.7 ;  
ARGIS INT.7 ;  
NCEND ...
```

8.1 Anweisungen zur direkten Tasksteuerung

Die Zustandswechsel werden ohne
Prioritätsänderung vorgenommen.

ändern

ACTIVATE-COMMAND:

'ACTI ' TASK-MODE ' '

[TASK-IDENTIFICATION].

Der Satz beginnend mit

PRIORITY.....

entfällt.

8.2 Anweisungen für die Taskzuteilung

Änderung von

SCHEDULE-1:

```
(CODE-ELEMENT EOL )$  
( 'AT CLOCK ' /  
'AFTER DUR ' ) [OPERAND] EOL /  
'WHEN '  
(IRPT-MODE ' ' G-CATEGORY ' '  
  XSYMB [ ' ( ) ' IRPT-MODE ] ) /  
(DOPE-MODE ' ' G-CATEGORY ' '  
  XSYMB  
  SINGLE-DIMENSION-SPECIFICATION  
) EOL.
```

Die letzten zwei Sätze des Abschnittes

Bei SCHEDULE-1 ...

werden ersetzt durch:

Felder von Interrupts lassen die Einplanungen zu:

- a) alle Elemente des Feldes
- b) ein beliebiges Element aus dem Feld.

8.3 Unterbrechungsoperationen

Ändern von

```
INTERRUPT-COMMAND:  
  ('ENABLE ' / 'DISABLE ' )  
  IRPT-MODE ' ' G-CATEGORY ' '  
  XSYMB ['(') ' IRPT-MODE ] EOL.
```

```
INDUCE-COMMAND:  
  'INDUCE ' SIGNAL-MODE ' '  
  G-CATEGORY ' '  
  XSYMB ['(') ' SIGNAL-MODE ] EOL.
```

Der Satz beginnend mit

VARIABLE ...

entfällt.

9. Anweisungen für die Ein-/Ausgabe

Es gibt drei Arten von Anweisungen für die Ein-/Ausgabe:

TRANSPUT:

DATION-MANAGEMENT-COMMANDS /
DATION-TRANSPUT-COMMANDS /
TRANSPUT-LISTS.

9.1 Verwaltung der Datenwege

Die Verwaltung beschränkt sich auf

DATION-MANAGEMENT-COMMANDS:
OPEN-COMMAND /
CLOSE-COMMAND.

Anweisungen zum Einrichten von Datenwegen sind nicht vorgesehen. Die entsprechenden Aufgaben werden durch besondere Funktionen in den FORMAT-Listen bewirkt (Abschnitt 9.4).

9.2 Öffnen und Schließen von DATIONS

Der Zugriff auf DATIONS, die vom Anwender definiert sind, muß durch folgende Anweisungen explizit synchronisiert werden:

OPEN-COMMAND:

```
'OPEN' DATION-OPERAND EOL /  
'OPENF' DATION-OPERAND EOL  
FORMAT-LIST-REFERNCE  
'IOEND '.
```

CLOSE-COMMAND:

```
'CLOSE' DATION-OPERAND EOL /  
'CLOSEF' DATION-OPERAND EOL  
FORMAT-LIST-REFERENCE  
'IOEND '.
```

wobei durch

FORMAT-LIST-REFERENCE:

```
'RFLIST ' LIST-HEADING EOL.
```

der Verweis auf die entsprechende FORMAT-LISTE erfolgt.

DATION-OPERAND:

```
' ' DATION-MODE ' ' VARIABLE.
```

9.3 Anweisungen zur Übertragung

Entsprechend dem Mode der zu übertragenden Darstellung wird die Ein- und Ausgabe jeweils durch einen speziellen Satz von Anweisungen vorgenommen.

INOUT-OPERAT:

'GET ' / 'PUT' /
'READ' / 'WRIT' /
'TAKE' / 'SEND'.

Der Befehlskörper wird mit Zusätzen versehen, um den weiteren Aufbau der Anweisungen anzudeuten:

D für Referenz auf DATA-LIST
F für Referenz auf FORMAT-LIST
DF für Referenzen auf bei Listen
(in dieser Reihenfolge).

Analog zur FORMAT-LISTE gilt:

DATA-LIST-REFERENCE:

'RDLIST ' LIST-HEADING EOL.

DATION-TRANSPUT-COMMANDS:

INOUT-D-COMMANDS /
INOUT-F-COMMANDS /
INOUT-DF-COMMANDS.

INOUT-D-COMMANDS:

INOUT-OPERAT 'D' DATION-OPERAND EOL
DATA-LIST-REFERENCE
'IOEND '.

INOUT-F-COMMANDS:

INOUT-OPERAT 'F' DATION-OPERAND EOL
FORMAT-LIST-REFERNCE
'IOEND '.

INOUT-DF-COMMANDS:

INOUT-OPERAT 'DF' DATION-OPERAND EOL
DATA-LIST-REFERENCE
FORMAT-LIST-REFERENCE
'IOEND '.

9.4 Format-Listen

In den Format-Listen sind die Operationen zusammengetragen, die in Ein-/Ausgabe-Anweisungen auf dem Control-Channel einer DATION wirken.

FORMAT-LIST:

```
[LOCAL-SYMBOL-IDENTIFICATION EOL]
'FLIST ' LIST-HEADING EOL
      (FORMAT-LIST-ELEMENT EOL )
'LISED ' LIST-HEADING.
```

wobei

LIST-HEADING:

```
' ' LIST-CATEGORY ' ' SYMBOL.
```

Der innere Aufbau

FORMAT-LIST-ELEMENT:

```
FORMAT-DENOTATION /
FORMAT-CALL       /
REPLICATION-BLOCK .
```

enthält die Formate als interpretierbare Liste oder als Prozeduraufruf.

C

Bei

FORMAT-DENOTATION:

'FELM FORMAT '

'F.' FORMAT-IDENTIFICATION

['(' INTEGER-LIST ')'].

liegen alle Formatparameter als Konstanten vor.

Andernfalls wird folgende Form erzeugt:

FOMRAT-CALL:

FORMAT-PRAGMAT-CALL /

FORMAT-PROC-CALL .

Alle nicht identifizierbaren Formatelemente werden als implementationsabhängige Zeichenfolge dargestellt.

FORMAT-PRAGMAT-CALL:

'FELM FORMAT ' 'X GPRAGM' EOL

('PRAGM ' PRAGMAT EOL)*

'FEND FORMAT ' 'X GPRAGM'.

FORMAT-PROC-CALL:

```
'FELM FORMAT ' 'X ' FORMAT-IDENTIFICATION EOL
( (CODE-ELEMENT EOL )$
'FARGIS '
(SINGLE-INT-MODE ' ' [OPERAND] )/
(SINGLE-CHARACTER-MODE ' ' OPERAND /
DOPE-MODE ' '
ARRAY-OR-SLICE-IDENTIFICATION)
EOL )•
'FEND FORMAT ' 'X ' FORMAT-IDENTIFICATION.
```

FORMAT-IDENTIFICATION:

```
'X' / 'SKIP' / 'PAGE' / 'ADV' /
'COL' / 'LINE' / 'POS' /
'A' / 'B' / 'B3' / 'B4' /
'D' / 'E' / 'F' / 'LIST' / 'T' /
'IDF' / 'OLD' / 'NEW' / 'ANY'.
```

Im Wiederholungsblock

REPLICATION-BLOCK:

```
'REP ' REP-LEVEL ' ' INTEGER-CONSTANT EOL  
  ( FORMAT-LIST-ELEMENT EOL )  
'REPED ' REP-LEVEL.
```

beschreibt

REP-LEVEL:

INTEGER.

die Tiefe der Schachtelung.

Der Wiederholungsfaktor wird durch INTEGER-CONSTANT angegeben.

Beispiel

OPEN-control aus BASIS-PEARL:

(IDF ('CIMIC/P'),NEW)

wird dargestellt:

SPACE STR.7 DISPO \$L99 S 67,73,77,73,47,80;

...

FLIST DISPO \$L100;

FELM FORMAT X IDF;

FARGIS STR.7 DISPO \$L99;

FEND FORMAT X IDF;

FELM FORMAT F NEW;

LISED DISPO \$L100;

Beispiel

Ein-/Ausgabe-Format aus BASIS-PEARL:

(POS (1,6), (3) (X(5),F(5,2)))

```
FLIST STATIC $L200;  
FELM FORMAT X POS;  
FARGIS INT.15 STATIC $L199;  
FARGIS INT.3 AINT 6;  
FEND FORMAT X POS;  
REP 1 AINT 3;  
FELM FORMAT F X(5);  
FELM FORMAT F F(5,2);  
REPED 1;  
LISED STATIC $L200;
```

9.5 Daten-Listen

In den Daten-Listen werden die Objekte zusammengetragen, die in Ein-/Ausgabe-Anweisungen auf den Data-Channel einer DATION wirken.

DATA-LIST:

```
'DLIST ' LIST-HEADING EOL  
  (DATA-LIST-ELEMENT EOL)•  
'LISED ' LIST-HEADING.
```

der innere Aufbau

DATA-LIST-ELEMENT:

```
( CODE-ELEMENT EOL )$  
'DELM ' ((SINGLE-SIMPLE-MODE /  
          STRUCTURE-MODE-IDENTIFICATION)  
          ' ' [OPERAND] ) /  
(DOPE-MODE ' '  
          ARRAY-OR-SLICE-IDENTIFICATION).
```

ist für Ein- und Ausgabe gleichartig.