

K.D. Reinartz  
Inst. f. Kristallographie  
Technische Hochschule  
Aachen

"Syntax von ROSA, Programm mit  
symbolischen Adressen.  
(PROSA-Untermenge)"

---

Syntax von ROSA, Programm mit symbolischen Adressen  
(PROSA - Untermenge)

K.D. Reinartz

Institut für Kristallographie, Techn. Hochschule  
Aachen

```
< Programm > := < Vorspann > < Datenfolge >;
< Datenfolge > := < leer > / < Datenfolge > < Date > 'WR'
< Date >      := 'ZWR' < Kommentar > / < Datenadresse > < Wort >
1  < Kommentar > := < leer > / < Kommentar > < Kommentarzeichen >
.1  < Kommentarzeichen > := < Buchstabe > / < Ziffer > / < sonstige
    Zeichen >
...1 < Buchstabe > := A/B/..../Z
...2 < Ziffer > := < Oktalziffer > /8/9
...1 < Oktalziffer > := 0/1/2/ ... /7
...3 < sonstige Zeichen > := < Symbolsonderzeichen > / < allgemeine
    Zeichen > /= / ) /[
...1 < Symbolsonderzeichen > := +/ -/x/ :/ 'ZWR'/'Wer da'/' ]/_10/_./,/'/(
...2 < allgemeine Zeichen > := 'ZL'/'BU'/'ZI'/'ZWR'
    < alle Zeichen > := < Buchstabe > / < Ziffer > / < sonstige Zei-
        chen > / 'WR' /;
2  < Datenadresse > := < leer > / ( < Symbol > )
.1  < Symbol > := < Buchstabe > / < Symbol > < Symbolzeichen >
...1 < Symbolzeichen > := < Buchstabe > / < Ziffer > / < Symbolsonder-
    zeichen >
3  < Wort > := < leer > / < OZ > / < DZ > / < GK > / < Befehl >
.1  < OZ > := OZ < Oktalzahl >
...1 < Oktalzahl > := < leer > / < Vorzeichen > < Oktalzahl ohne Vor-
    zeichen >
...1 < Vorzeichen > := < Plus > / -
....1 < Plus > := < leer > / +
...2 < Oktalzahl ohne Vorzeichen > := < Oktalziffer > / < Oktalzahl
    ohne Vorzeichen > < Oktalziffer >
.2  < DZ > := DZ < Dezimalzahl >
...1 < Dezimalzahl > := < leer > / < Vorzeichen > < Dezimalzahl ohne
    Vorzeichen >
...1 < Dezimalzahl ohne Vorzeichen > := < Dezimalziffer > / < Dezimal-
    zahl ohne Vorzeichen > < Dezimalziffer >
```

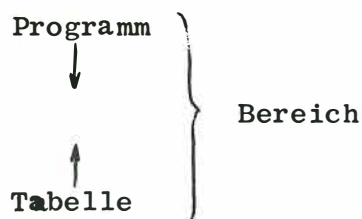
- ```

.3      < GK > := GK < Gleitkommazahl >
..1     < Gleitkommazahl > := < leer > / < Vorzeichen > < Dezimalzahl ohne
   Vorzeichen > < Exponententeil >
...1    < Exponententeil > := < leer > /10 < Vorzeichen > < Dezimalzahl
   ohne Vorzeichen >
.4      < Befehl > := < Operationsteil > < Kennzeichen > < Operandenadres-
   se >
..1     < Operationsteil > := ADD/EAO/.../EA7/EPR/EVO/.../EV7/KPL/MLT/NOP/
   ODR/SAM/SAP/SGN/SKU/SPR/STP/SUB/SUL/SUM/TAS/
   TEL/TEM/TEP/UGL/UND/UNT/VAL/VAR/VDL/VDR/VLL/
   VLR/VSE
..2     < Kennzeichen > := < Akkukennzeichen > < Substitution > < Unter-
   brechbarkeit >
...1    < Akkukennzeichen > := < leer > / '
...2    < Substitution > := < leer > / :
...3    < Unterbrechbarkeit > := < leer > / .
..3     < Operandenadresse > := < leer > / < Oktalzahl ohne Vorzeichen > /
   (< Unterprogrammnummer > < Platznummer >)/
   < Datenadresse > < Oktalzahl > /x <Oktal-
   zahl >
...1    < Unterprogrammnummer > := < Oktalzahl ohne Vorzeichen >
...2    < Platznummer > := < leer > / , < Oktalzahl ohne Vorzeichen >

```

# Erläuterungen zum Assembler

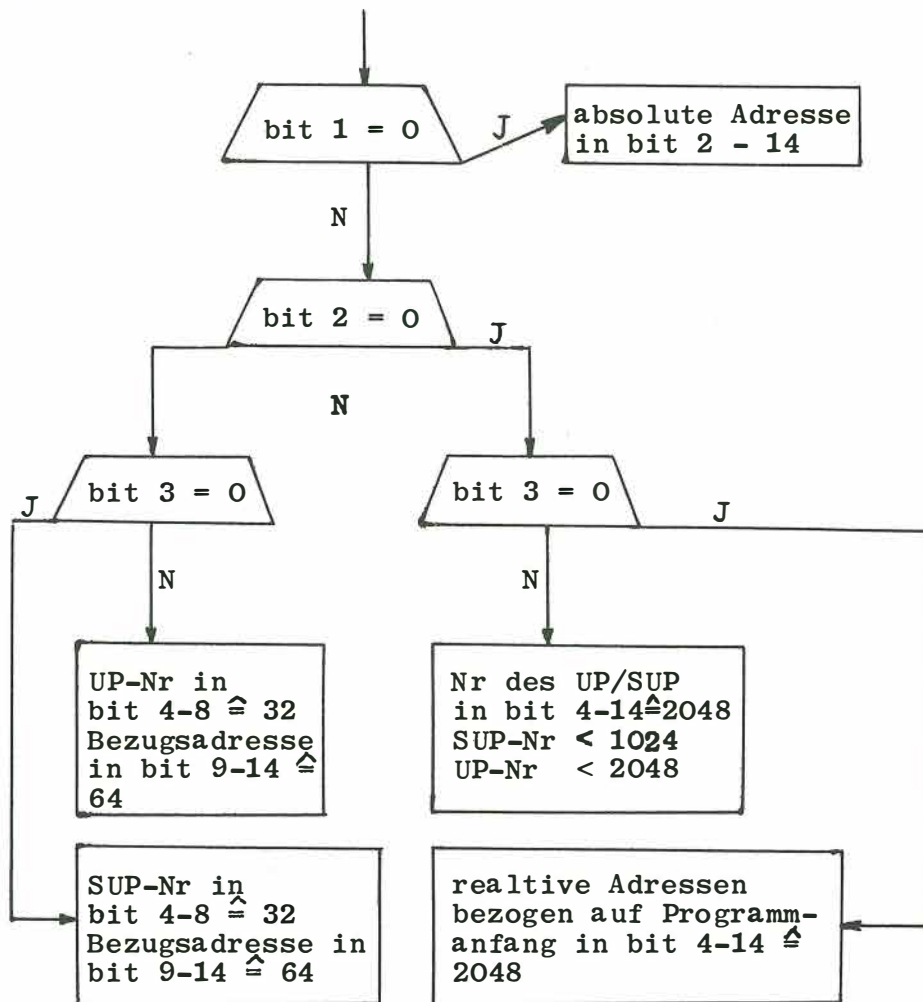
1.  $\langle \text{Vorspann} \rangle := \text{'WR' 'ZL' } \langle \text{Nr} \rangle : \text{'WR' 'ZL'}$   
 $\text{DZ} : \langle \text{Anfangsadr.} \rangle - \text{DZ} : \langle \text{Endadresse} \rangle - \text{DZ} :$   
 $\langle \text{P-Nr} \rangle$   
 $\langle \text{Nr} \rangle$  ist die Nummer des assemblierenden Hauptprogramms  
 $\langle \text{Anfangsadr.} \rangle,$   
 $\langle \text{Endadresse} \rangle$  geben den Bereich an, in dem das zu assemblierende Programm einschliesslich seiner Symboltabelle aufgebaut wird



< P-Nr > Hauptprogramm - Nr des zu assemblierenden Programms.  
Ist das Programm ein UP/SUP, so ist die P-Nr = 0.

## 2. Programmaufbau

- 1) Die erste Zelle des Programms muss als erstes < Wort >  
NOP < Nummer > enthalten  
< Nummer > = 0 bei Hauptprogramm  
≠ 0 bei UP/SUP
- 2) Die verschiedenen Operandenformen und ihre Darstellung im Maschinewort



- 3) Beim letzten relativen Befehl eines Programms ist die Markierungs-  
stelle besetzt.  
Dadurch entsteht ein voll relatives Programm, das über die

Programmnummern mit anderen UP/SUP verbunden ist. Das Organisationsprogramm baut beim Laden der einzelnen Programme die Nummern ab, so dass eine relative Form ohne Nummern-Verbindungen entsteht. Erst bei Start wird diese relative Form aufgehoben.

4) Die Eingabe und Ausgabe eines Zeichens erfolgt im Assembler durch je einen UNT-Aufruf.

5) Fehlermeldungen erfolgen on-line in der Form 'ZWR' F < Fehler-nummer > 'WR' 'ZL'

- F1 < Operationsteil > oder Zahltyp falsch
- F2 < Gleitkommazahl > enthält unzulässige Zeichen
- F3 < Befehl > " " "
- F4 < Symbol > beginnt nicht mit einem Buchstaben
- F5 Programm + Symbolliste werden zu land  
(Ende des Programms)
- F6 < Symbol > mehrfach definiert  
(erstes Symbol gilt)
- F8 Das Programm besitzt mehr als 512 Symbole  
(weitere Befehle mit < Datenadresse > ≠ < leer > werden  
überlesen)
- F9 Symbole nicht alle definiert  
(Ende des Programms)
- F10 Dezimalzahl enthält mehr als 8 Ziffern

Normalerweise kann nach einer Fehlermeldung die fehlerhafte < Date > neu eingegeben werden, da sie nicht abgespeichert wird.

Das Irrungszeichen [ lässt alle vorhergehenden Zeichen der < Date > übergehen.

6) Gleitkommazahlen werden in zwei aufeinanderfolgenden Worten dargestellt.

- 1. Wort Mantisse bit 1-24
- 2. Wort dualer Exponent bit 1-14

3. Der Assembler ist eingebettet in ein eigenes Betriebssystem für einen Kernspeicher mit maximal 8K Worten : Rechner z.B. 303 P.