

**Dr. H. Mittendorf**

Siemens AG Karlsruhe

FORTRAN IV H-Level für das Siemens-System 300

## FORTTRAN IV H-Level für das Siemens-System 300

*Die große Anzahl der verschiedenen Aufgaben, die heute von Prozeßrechnern gelöst werden müssen, macht in zunehmendem Maße die Verwendung höherer Programmiersprachen erforderlich. Hierbei nimmt die Sprache FORTTRAN zur Zeit eine dominierende Stellung ein; für die Aufgaben der Prozeßprogrammierung sind überdies zusätzliche Sprachelemente vorgeschlagen, die unter dem Sammelbegriff Prozeß-FORTTRAN in die Literatur eingegangen sind. Dem Zuge dieser Entwicklung folgend entwickelt unser Haus daher einen neuen FORTTRAN-Compiler, der diesen neuen Anforderungen Rechnung trägt.*

Compiler für die älteren Maschinen des 300-Systems mußten noch mit einer minimalen Anlagenausstattung (z. B. nur LSK als Ein-Ausgabegerät, kein Externspeicher) arbeiten können. Sinnvoll war dies nur möglich, wenn man im Falle von FORTTRAN auch den Sprachumfang beschränkte; daher wurde zunächst der E-Level zugrunde gelegt. Erst mit den erweiterten Mitteln der DVA 306 war es gerechtfertigt, den vollen H-Level zu implementieren.

### 1. Eigenschaften der Sprache

Mit FORTTRAN 306 ist der Sprachumfang von FORTTRAN bezeichnet, den der zur Zeit in Arbeit befindliche Übersetzer für das SIEMENS SYSTEM 300 bietet. Er ist zu unterscheiden von der bisher unter dem Namen FORTTRAN 300 verfügbaren Version. Ungeachtet dieser (vorläufigen) Bezeichnung können jedoch sowohl der Übersetzer als auch die von ihm erzeugten Programme auf den Anlagen 301 bis 305 ablaufen, soweit Anlagenausstattung, Speicherplatz und verdrahteter oder simulierter Befehlsvorrat hierfür ausreichen.

FORTTRAN 306 umfaßt den vollen Sprachumfang, wie er von der American Standards Association (ASA) genormt wurde. Darüber hinaus sind erlaubt:

- Implizite Spezifikationsanweisung (IMPLICIT)
- Dateninitialisierung bei expliziter Spezifikationsanweisung
- arithmetische Ausdrücke gemischten Typs
- Längenangabe von Variablen mit \*2, \*4, \*8 und \*16
- Komplexe Rechnung mit doppelter Genauigkeit (COMPLEX\*16)
- Felder mit mehr als 3 (7) Dimensionen
- T-Format
- NAMelist-Anweisung
- Apostroph als Begrenzungszeichen von Literalkonstanten ENTRY-ANWEISUNG bei Unterprogrammen

FORTTRAN 306 wurde, soweit möglich, an das FORTTRAN des SIEMENS SYSTEM 4004 angelehnt. Da es sich bei allen Anlagen dieses Systems um Byte-Rechner handelt, beim SYSTEM 300 dagegen um Wort-Rechner mit 24 bit-Wortlänge, ergeben sich einige Abweichungen, die im folgenden beschrieben werden sollen.

Der Zeichenvorrat basiert auf dem 6-bit-Code des SYSTEM 300 und ist von dem 8-bit-Code der Rechner des SYSTEM 4004 verschieden. Er ist jedoch identisch mit dem der IBM 7090 und 7094 und kann daher von einer SIEMENS 4004-Anlage mit dem sog. 7090/7094-Zusatz verarbeitet werden. Umgekehrt kann der 306-Übersetzer auch Programme übersetzen, die im Lochkartencode für das System 4004 vorliegen, sofern sie folgenden Zeichenvorrat nicht überschreiten:

Alle anderen Zeichen des Interncodes dürfen nur in Literalen vorkommen. \$ ist ein Sonderzeichen, & ist nicht erlaubt.

Zwischen den Bytes des SYSTEM 4004 und den Festworten des SYSTEM 300 wird folgende Zuordnung festgelegt:

Damit ergeben sich folgende Datentypen:

Bei Literalen werden 4 Zeichen je Wort gespeichert.  
Zu beachten ist die unterschiedliche Behandlung des Typs LOGICAL\*1, da Halbworte nicht adressierbar sind.

Die Gleichsetzung von 2 Byte mit einem Festwort ergibt nur eine formale Übereinstimmung. Es besteht jedoch eine Abweichung im Zahlenbereich der Variablen:

## 2. Beschreibung und Handhabung des Übersetzers

90

Der eigentliche Übersetzer besteht aus 11 Pässen und benötigt mindestens 4096 FW im Arbeitsspeicher. Für automatisches Binden und für Ausgabe des erzeugten Programms in Assemblersprache PROSA 300 kommen jeweils 2 weitere Pässe hinzu. Zur Beschleunigung des Übersetzungsvorganges kann beim Laden des Systems ein größerer Bereich im Arbeitsspeicher (bis 16 KFW) zugeteilt werden.

Auf dem Externspeicher benötigen Übersetzer und Binder einschließlich der erforderlichen Systembibliothek(en) über 200 KFW. Durch Verzicht auf nicht benötigte Bibliotheksfunktionen läßt sich der Externspeicher-Platzbedarf verringern.

### *2. 1 Laden des Übersetzers*

Der Übersetzer wird mit einem besonderen Ladeprogramm in die Arbeitsform vom Externspeicher geladen. Dabei werden seine Bereitstelladresse und sein Laufbereich festgelegt.

### *2. 2 Übersetzungslauf*

Zur Steuerung des Übersetzungsvorganges stehen wahlweise zwei Steuerprogramme zur Verfügung. Mit deren Hilfe wird die benötigte Information entweder über einen Bedienungsblattschreiber oder nach Art eines Monitors über Steuerlochkarten eingegeben. Folgende Möglichkeiten stehen zur Verfügung:

- a) Protokollierung auf Blattschreiber oder Schnelldrucker. Es werden alle Bedienungsanweisungen mitprotokolliert. Fehlermeldungen erscheinen als ausführlicher, selbsterklärender Text im Protokoll des Quellprogramms unmittelbar nach der fehlerhaften Zeile.
  - b) Eingabe des Quellprogrammes von Lochkarten (BCD-Code = Normalfall oder EBCDIC-Code wie beim SYSTEM 4004), von Lochstreifen oder aus einer wählbaren Plattenspeicher-Datei.
  - c) Ausgabe des erzeugten Programms als Modul oder als fertig gebundenes, selbständig lauffähiges Programm.
  - d) PROSA-Ausgabe des erzeugten Moduls. Im PROSA-Protokoll erscheinen die FORTRAN-Namen des Quellprogramms.
  - e) Einbau von Testhilfen:
    - Indextrace      Überwachung der Indizes von Feldkomponenten
    - Valuetrace      Protokollierung aller Wertzuweisungen beliebig wählbarer Variablen.
    - Sprungtrace    Protokollierung aller Statement-Nummern, die durch Sprung – also nicht im sequentiellen Programmablauf – erreicht werden.
    - Zeilen-trace    Fehlermeldungen zur Programmlaufzeit werden als ausführlicher, selbsterklärender Text mit Modulnamen und Zeilennummer ausgegeben.
  - f) Zusätzliche Protokolle, wie Namensliste, Markenliste und Dimensionsliste.
  - g) Verlängern der Arbeitsdateien, falls deren Standardlänge nicht ausreicht.
  - h) Löschen der Arbeitsdateien nach der Übersetzung, um Externspeicherplatz freizugeben.
- Zwingend notwendig sind nur Angaben über die Protokollierung; alle anderen Wahlmöglichkeiten werden entweder nicht aktiviert oder durch Standardmaßnahmen ersetzt.

## *3. Erzeugung der gebundenen Programme*

Für die Herstellung ablauffähiger gebundener Objekt-Programme (in MC) sind nach der Übersetzung noch zusätzliche Maßnahmen erforderlich.

### *3. 1. Binden der erzeugten Module*

Der Binder erzeugt ablauffähige Programme aus Modulen, die teils vom Übersetzer erzeugt sind, teils in Standardbibliotheken vorliegen. Solche Module können auch aus PROSA-Programmen mit Hilfe des PROSA-Übersetzers erzeugt sein.

Durch Steuerkarten kann festgelegt werden

- a) welche Bibliotheken in welcher Reihenfolge durchsucht werden sollen,
- b) welche Moduln aus welchen Bibliotheken verwendet bzw. von der Verwendung ausgeschlossen werden sollen,
- c) bei welchem Adreßpegel ein jeweils anzugebender COMMON-Bereich einzubauen ist.

Zusätzlich für den Overlay-Binder:

- d) welche Segmente überlagert werden sollen (Baumstruktur)
- e) wo ein neuer Verzweigungsbaum beginnt (Region)

Ein Beispiel für eine Overlay-Struktur zeigt Abb. 2.

Die Bindestrategie ermöglicht die Herstellung verschiedener Programmvarianten — z. B. auch für verschiedene Betriebssysteme — ohne Neuübersetzung des Quellprogramms, da der Übersetzer nie unmittelbar Betriebssystem-Aufrufe in Programmen erzeugt, sondern diese stets in den einzubindenden Moduln der Standardbibliotheken stehen.

### 3. 2. *Mischen von FORTRAN- und PROSA-Programmteilen*

Die Moduln, aus denen ein Gesamtprogramm zusammengestellt wird, können selbstverständlich auch in PROSA geschrieben sein, wie z. B. die Standardfunktionen. Ein solcher Modul kann ein Hauptprogramm, eine SUBROUTINE oder eine FUNCTION sein. Es müssen lediglich bestimmte Normierungen bei der Parameterversorgung von Unterprogrammen, bei der Verwendung von COMMON-Bereichen und BLOCK DATA-Unterprogrammen eingehalten werden.

### 3. 3. *Programmkopplung*

Unter Programmkopplung sei hier die Zusammenarbeit mit anderen, in der gleichen Anlage simultanlaufenden Programmen verstanden. Hier bestehen folgende Möglichkeiten:

- a) Datenaustausch über Arbeitsspeicher oder Externspeicher mit Hilfe von READ/WRITE, binär oder alphanumerisch, außerdem im Arbeitsspeicher ohne Umspeichern über COMMON-Bereiche.
- b) Koordination durch Organisationsaufrufe an das Betriebssystem mit Hilfe entsprechender CALL-Statements.

## 4. *Prozeß-FORTRAN*

### 4. 1. *Grundsätzliches*

Prozeß-FORTRAN ist eine in der Literatur gebräuchliche Bezeichnung für die Erweiterung des Sprachvorrates von FORTRAN zur Programmierung von Prozeßaufgaben. Bei dem von uns verfolgten Weg wird das vorliegende Konzept des H-Level ausgebaut. Die vorgeschlagenen Erweiterungen betreffen neue Verknüpfungsmöglichkeiten zwischen Variablen, besondere Anweisungen für Textverarbeitung, zusätzliche Standardfunktionen zur Programmkoordination und für Prozeß-Ein/Ausgaben (PURDUE-Norm). Außerdem sollen wichtige Standardfunktionen von mehreren Programmen benutzbar sein. Bei Einhaltung bestimmter Normierungen muß die Einfügung von PROSA-Zeilen in das FORTRAN-Programm erlaubt sein.

Im folgenden werden die neuen Sprachmittel kurz vorgestellt.

### 4. 2. *Neue Verknüpfungen zwischen Variablen*

Zunächst wird die syntaktische Trennung von LOGICAL\*2 und INTEGER\*2 (ebenso für die \*4-Größen) aufgehoben.

Damit können nun .AND., .OR. usw. auch auf INTEGER-Variable und umgekehrt \*, +, / usw. auch auf logische Variable angewendet werden.

Zusätzlich werden für diese Variablen neue Operatoren eingeführt:

- .EOR.        exklusives Oder
- .ISHIFT.K   arithmetische Verschiebung
- .LSHIFT.K   logische Verschiebung
- K > 0 bedeutet Verschiebung nach links
- K < 0 bedeutet Verschiebung nach rechts

#### 4. 3. *Neue Hilfsmittel für Textverarbeitung*

Um formatgesteuerte Texte auch im Kernspeicher auf- und abbauen zu können, werden entsprechend READ und WRITE die Anweisungen

EN-CODE        (K, F-Nr., Feld) Liste  
DE-CODE

eingeführt. Damit können K Zeichen ( $K \leq 120$ ) in das Feld „Feld“ geschrieben (EN CODE) bzw. aus ihm gelesen werden (DE-CODE). Mit diesen Funktionen ist eine flexible Textverarbeitung möglich.

Mit zusätzlichen Formatschlüsseln läßt sich das Datum in der Form WT.TT.MM.JJ (z. B. MI. 26.05.71) und die Uhrzeit in der Form HH.MM.SS ausgeben (z. B. 07.15.30).

#### 4. 4. *Execute-Aufrufe*

Zur Kennzeichnung von Unterprogrammen, die Sonderfunktionen des Betriebssystems ausführen, wird ein neues Schlüsselwort eingeführt:

EXECUTE Name (Parameterliste)

Diese Anweisung ist formal mit der Anweisung CALL vergleichbar. Jedoch werden durch das neue Schlüsselwort Doppeldeutigkeiten vermieden, die auftreten würden, wenn ein Anwender eigene Unterprogramme mit demselben Namen belegt. Bedeutet etwa EXECUTE P START (N) den Start eines Hauptprogramms der Priorität N, so kann unabhängig davon ein Anwender ein eigenes Unterprogramm mit CALL P START (N) aufrufen.

Man vermeidet so die Notwendigkeit, eine Reihe von reservierten Namen erfinden zu müssen.

#### 4. 5. *Standardfunktionen für Prozeßanwendungen*

Das „Interim Procedural Language Committee“ des PURDUE-Workshop (Lafayette, Indiana) hat im Oktober 1970 eine Reihe von Unterprogrammen für Prozeßanwendungen zur Normierung vorgeschlagen. Wir werden diese Programme zur Verfügung stellen und aus den schon erwähnten Gründen über EXECUTE aufrufbar machen. Insbesondere gehören hierzu

Starten und Beenden von Programmen nach einer Zeitverzögerung sowie die zeitverzögerte Fortsetzung des aufrufenden Programms.

Bei den Prozeß-Ein/Ausgaben sind zur Normierung vorgeschlagen

Analogeingabe sequentiell,  
Analogeingabe wahlfrei,  
Digitaleingabe – Impulsausgabe  
(alle Funktionen mit und ohne Warteaufruf).

Bestimmte Standardfunktionen werden unabhängig vom aufrufenden Programm in den sogenannten Common-Code-Bereich eingeordnet. Sie besitzen dann reentrant-Eigenschaften.

#### 4. 6. *Einfügen von PROSA-Zeilen in FORTRAN-Programme*

Für die Kopplung von FORTRAN- und PROSA-Programmen gibt es außer der bekannten Möglichkeit über Unterprogramme auch noch das direkte Einfügen von PROSA-Zeilen in das FORTRAN-Quellprogramm. Diese PROSA-Anweisungsfolge beginnt mit dem Schlüsselwort PROSA und wird jeweils durch das Schlüsselwort FORTRAN abgeschlossen.

Für das PROSA-Programm gibt es jedoch einige Einschränkungen:

- a) Es sind nur *lokale* Adreßdefinitionen erlaubt.
- b) Im Operationsteil dürfen nur Befehlsschlüssel stehen (nicht GK, DZ, AP usw.).
- c) Es sind im Adreßteil nur lokale Adressen, Variablennamen des FORTRAN-Programms, Externadressen und Konstante erlaubt.

Diese Einschränkungen sind notwendig, um eine sinnvolle Syntax-Kontrolle auch in den PROSA-Programmteilen durchzuführen.

Mit diesen Hilfsmitteln einer Prozeßversion des FORTRAN IV stehen, so meinen wir, dem Anwender jetzt sehr viel erweiterte Programmierungsmöglichkeiten zur Verfügung. Wir hoffen, daß sie nach Fertigstellung des Compilers auch ausgenutzt werden und somit das Programmieren von Prozeßaufgaben in höheren Sprachen sich weiter durchsetzt.

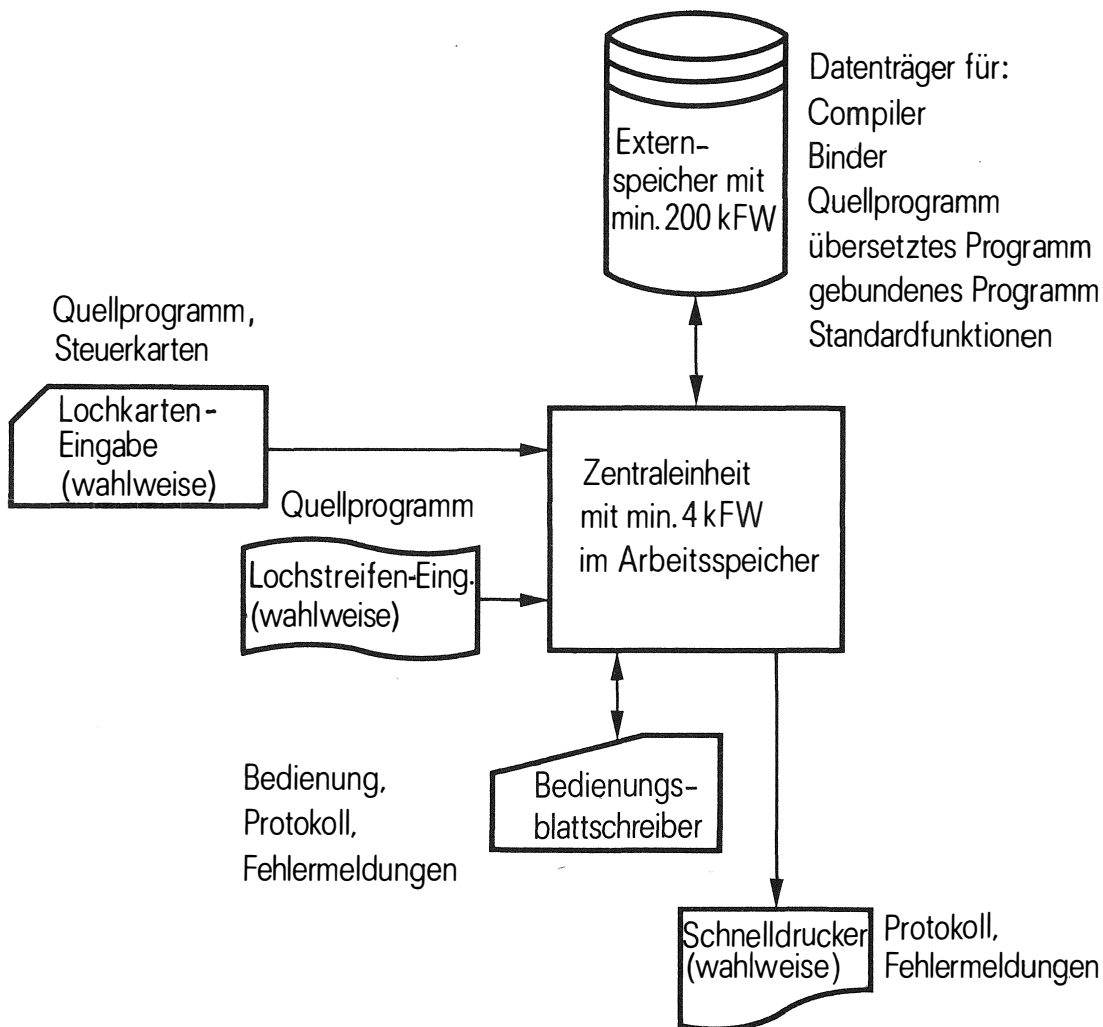
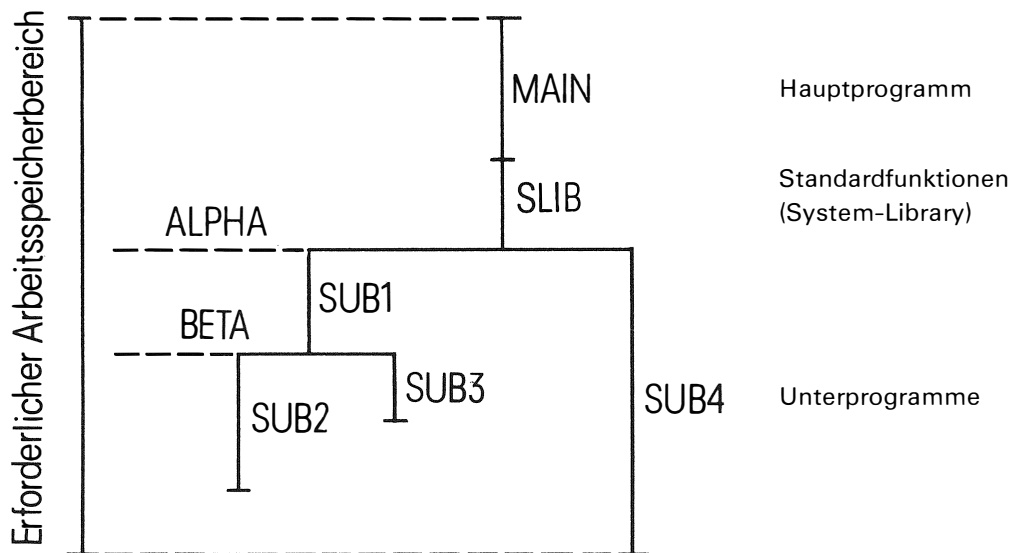


Abb. 1 Anlagenkonfiguration für FORTRAN IV-Level



*Steuerkarten für diese Struktur*

PROGRAM	TEST, 2000	Programm TEST soll ab 2000 bereitgestellt werden
INCLUDE	MAIN	System-Library an MAIN angekoppelt
LIBRARY	SLIB	
OVERLAY	ALPHA	Definition von ALPHA für den
INCLUDE	SUB1	
OVERLAY	BETA	
INCLUDE	SUB2	1. Hauptzweig
OVERLAY	BETA	
INCLUDE	SUB3	
OVERLAY	ALPHA	Definition von ALPHA für den
INCLUDE	SUB4	2. Hauptzweig
ENTRY		Ende der Steuerkarten

Abb. 2 Beispiel einer einfachen Overlay-Struktur