rc 4000 COMPUTER®

PRELIMINARY SPECIFICATION

FORTRAN

This specification – which is preliminary and may be subject to change – describes the Fortran compiler developed for the extended RC 4000 computer with drum and/or disc as backing store. The compiler translates a Fortran source program into a binary object program.

Program Configuration

The compiler and the object programs can only be executed under the control of the time-sharing system Monitor 1.

The compiler and the object programs can be loaded from the backing store by the basic operating system in the monitor. During execution input/output, storage protection, and program interruption are controlled via the monitor.

Fortran programs can be translated simultaneously by copies of the compiler loaded from the same area on the backing store into different storage areas.

Source Language

The source language is the ISO Fortran of the first level as described in "Draft ISO Recommendation No. 1539," ISO/TC 97 (CS-99) 153 E, with the exception of DATA initialization of local variables and logical unit numbers. These numbers are replaced by names, which facilitates the use of peripheral devices of all kinds.

Generalizations and extensions of the language include: mixing of types, in arithmetic expressions; general expressions, as subscrips and in DO statements, which makes it possible to count backwards; long integers, text constants, and bit patterns; mask operations; DATA initialization of common variables anywhere; multiple entries in procedures; optional inclusion of test program lines.

The Fortran program text can be input to the compiler from typewriter, paper tape, punched cards, magnetic tape, drum, disc, or a combination of these. Paper tape must be one-inch tape with 8 tracks punched in either the RC Flexowriter code or the ISO 7-bit character code. Punched cards must be 12-row, 80-column cards punched in the Hollerith or the ISO code. Input from drum, disc, or magnetic tape must be represented in the ISO 7-bit character code with 3 characters per word.

Procedures can be compiled separately, and are then available to all users like a standard procedure. As this also applies to Algol procedures, it is possible to use Algol procedures in Fortran programs and vice versa.

Object Program

The object program is stored on drum or disc as a sequence of relocatable segments of 256 words each. The program includes the automatic administration of transfers of segments to the core store at run time.

Local variables are set up dynamically. Variables of the logical type are stored in bytes, variables of the integer type in words, variables of the real type and long integers in double words, and double-precision and complex variables in two double words. Buffer areas for peripheral devices are declared at entry, and require core store space as specified by the programmer.

The checking of parameters in procedure calls and of labels in assigned GO TO statements is standard, whereas both the check against subscript bounds and the checking of common variable types is optional.

The execution time of a program depends on the amount of core store space available. The minimum requirement is 1,700 words plus the necessary space for variables. For long programs 4,000 words plus space for variables is recommended.

Typical Execution Times in Microseconds

reference to subscripted variable, 1 subscript	16
reference with check against bounds	25
integer + integer, local	7
integer + integer, common	10
long + long, local	12
real + real, local	18
double + double, local	240
GO TO statement, simple	7
DO statement, constant control, each loop	30
call of empty procedure, no parameters	50
call of empty procedure, 1 simple parameter	61
reference to simple parameter	5
call of sin, cos	450

Translation

The compiler requires a core store area of 4 K words and a working area on the backing store of sufficient size to hold the object program. If the backing store is a disc, a core store area of 8 K is recommended in order to obtain maximum compilation speed.

It is possible to translate Fortran procedures separately on the backing store and use them as external procedures within other Fortran or Algol programs. The procedure code is then copied into the object program during translation.

The compiler itself is written in the assembly language Slang 3.

rc 4000 COMPUTER®