

10. OPERATION

10.1. SOURCE CARD FORMAT

The source language statements to the compiler must come initially from punched cards. Only columns 1–72 are read for information (in free format) and following column 72 is considered to be a (space) delimiter. Columns 73–80 can be used for any purpose desired, e.g., short comments or serial identification. There is no restriction on placing statements on a card but the usual practice is to arrange them for easy reading and modification. The full 80 columns may be utilized for input data at execution time.

10.2. OPERATING INSTRUCTIONS

The ALGOL compiler operates like all the processors in the UNIVAC 1108 Executive System and, besides the standard options, includes some unique to itself. The available options are:

- A Accept the results of compilation even if errors were detected.
- B Inhibit the printing of block diagnostics. Without this option each statement that begins a block is preceded by the message

BLOCK XX LEVEL YY

where xx is a serial block number and yy is its static depth. Each statement ending a block is preceded by

END BLOCK XX

In addition each **BEGIN-END** pair is tagged with the message
Bnn and Enn

where nn is a serial count of the pairs. The values of nn and xx are not limited to a maximum of 99 but only 2 digits are printed.

- S Single spaced listing of ALGOL source statements.
- L The compiled assembly language instructions are listed along with the source code.
- N (or lack of any other print option) Suppress all printing by the processor. If 'N', disregard any other print option.
- T Print the timing for phases 1 and 2 of compilation.
- X Abort the run immediately if any error is found.
- Z Delete the formation of run-time diagnostic information.
- O,R References to subscripted variables normally generate a call to a library procedure. This procedure, besides calculating the proper address, also checks that the requested operation is legal (i.e., that the subscript variables are in the range of the declaration). With the R option (remove checking) this checking is not done but the address calculation is, thus giving greater speed to the object program. The O option (open) is even faster in that the necessary coding to calculate the address is compiled in line, thus removing the call and return times from the reference. Of course, the O option requires more main storage. The suggested use of these options is this: neither should be used when the program is being debugged. When the program is working, and if the subscript expressions are not data-dependent, then the R option should be used. If main storage permits, the O option should be used.