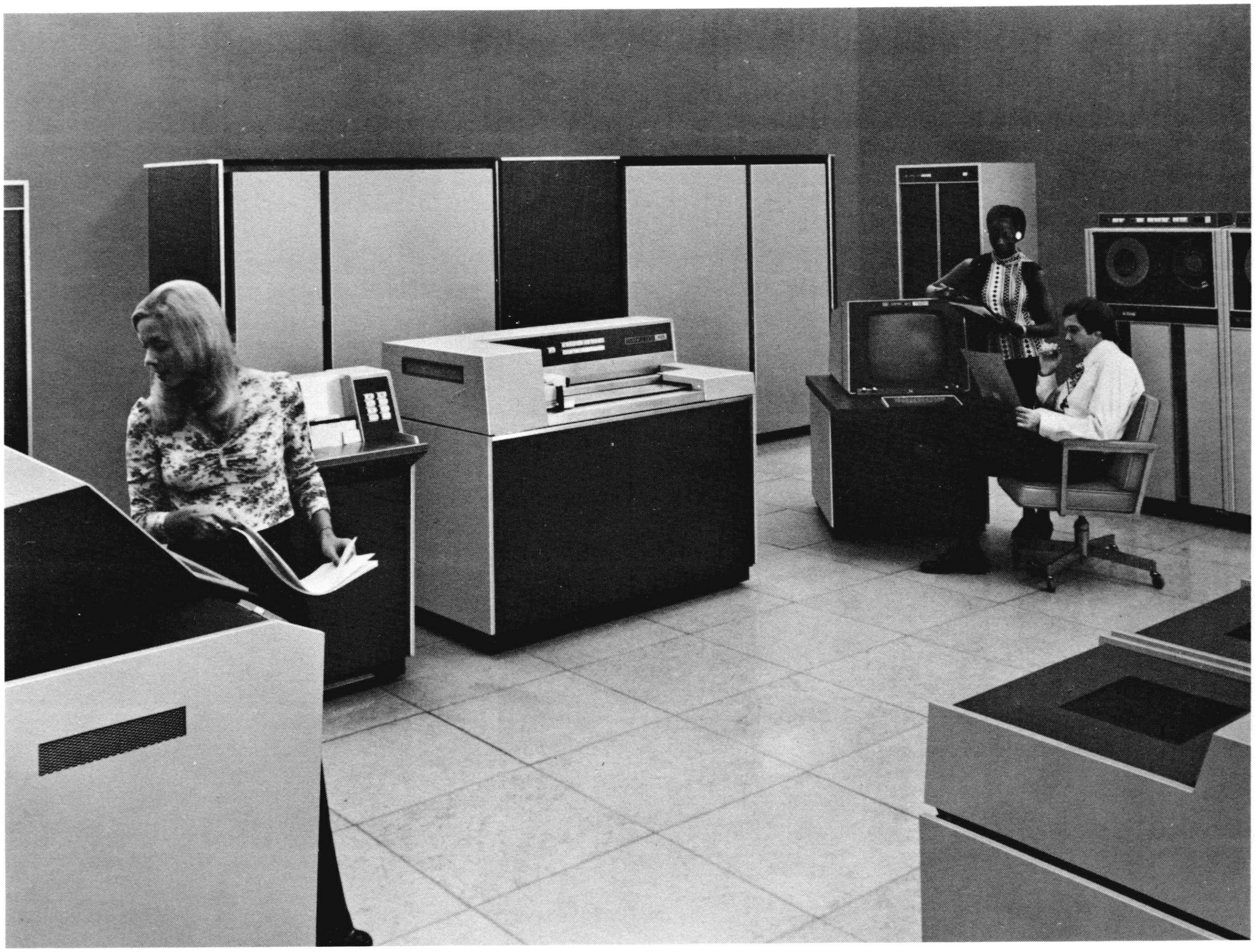
CONTROL DATA® CYBER 170 SERIES FORTRAN EXTENDED COMPILER

CONTROL DATA
CORPORATION



The FORTRAN Extended Compiler for the CDC® CYBER 170 Series Computer System provides for expressing mathematical and scientific problems in a familiar mathematical notation. The system is designed to provide the complete capability of the FORTRAN language and several additional features provide considerable power and flexibility.

FORTRAN Extended functions under control of the operating system of the computer. The object code produced by the compiler is comparable with code written by an experienced COMPASS programmer. Sophisticated techniques, such as in-line evaluation of some functions and critical path analysis of instruction sequences, are used to produce highly optimized object jobs which take full advantage of the speed and computational power of the CDC CYBER 170 system.

Debugging aids implemented with FORTRAN Extended provide a variety of features to aid in program checkout. Typical compiler output consists of a source code listing which includes easily understood diagnostics listed by statement numbers. The programmer can also obtain a cross-reference table of symbols used of a complete listing of COMPASS code generated by source statements.

FORTRAN Extended is a two-pass compiler. The first pass, using source language as an input, creates pseudo instructions for execution and necessary linkage data. The second pass uses these newly created instructions as input and optimizes them to create output in assembly language and macro format. This output is then used as input to the COMPASS Assembler.

During the first two passes, memory is used efficiently, since, as each compiler function is completed, the central memory space it used as program space becomes available as additional working storage.

Some of the features of the FORTRAN Extended system include:

- Use of OCTAL and HOLLERITH data in language elements.
- Use of MASKING expressions statements.
- Use of MASKING and MULTIPLE assignment statements.
- Two branch ARITHMETIC and LOGICAL IF.
- Use of IMPLICIT and LEVEL specification statement.
- Various utility subprograms, such as DUMP, TRACE.
- PRINT, PUNCH, BUFFER IN, BUFFER OUT, NAMELIST, ENCODE, and DECODE input/output statements.

- Comprehensive diagnostics.
- Flexible subroutine structure
  - Multiple entry points to subroutines
  - Mixed FORTRAN and COMPASS coded routines
  - Return to one of several statements in the main program through multiple statements.
- Use of overlays to reduce the amount of storage required.

Optimization techniques of FORTRAN Extended include:

- Elimination of redundant operations.
- Evaluation of array element addresses by the index function method.
- Critical path analysis of instruction sequences to maximize parallel operation.
- Reformation of subexpressions at compile time as opposed to execute time.
- Determination at compile time whether each reference to a formal parameter should be referred to by address substitution or by indirect addressing.
- Elimination of common remote parameter lists.
- Formation of simple constants by sets rather than by loads.
- Elimination of branches to the next instruction.
- In-line evaluation of some functions.
- Presetting arrays with a constant pattern handled at load time to eliminate generation of large binary decks.
- Special attention to DO loop processing to include indexing methods, array addressing and loop counting.

FORTRAN Extended is modular in structure and contains an extensive offering of programmer-defined macros and micros, so that it can be easily modified by users. Statements can be added quickly and easily, using a well-defined set of rules.

201,009 LITHO IN U.S.A. 3/74