EASYCODER FOR THE TAPE-RESIDENT MOD 1 OPERATING SYSTEM

SERIES 200

Easycoder, a powerful data processing tool for the Series 200 computers, is designed to provide a maximum of flexibility and power in programming while retaining simplicity in manipulation. With Easycoder, the user can easily and efficiently prepare a wide variety of programs for operation.

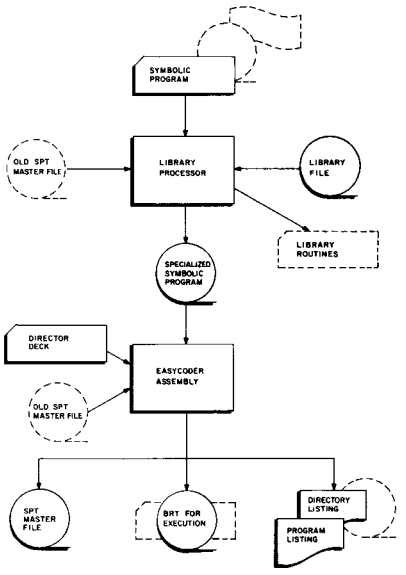
There are two versions of Easycoder for the Tape-Resident Mod 1 Operating System. Each version consists of three elements — a symbolic language, a library processor, and an assembler. Source programs written in the symbolic language are supplemented with precoded symbolic routines by the library processor and assembled into machine-language object programs by the assembler. The Easycoder symbolic language incorporates several types of symbolic statements, using easily remembered mnemonic operation codes and symbolic tags to facilitate actual coding of programs. The library processors permit the user to retrieve routines from a library of already prepared coding by simply issuing a call to the library, thereby eliminating repetitious steps in program preparation. The assemblers, which perform the actual translation from source to machine language, take full advantage of large tape-storage configurations to provide speed in operation. Various file updating functions can also be performed by the assemblers.

The standard version of Easycoder for the Mod 1 environment is Easycoder C. For installations where additional memory capacity and the corresponding hardware features are available, Easycoder D allows the use of floating-point instructions and storage protection. The following information pertains to both versions unless otherwise noted.

EASYCODER SYMBOLIC LANGUAGE

The Easycoder symbolic language incorporates three types of symbolic statements used in writing an Easycoder program — data formatting, assembly control, and data processing. Data formatting statements enable the programmer to reserve and punctuate memory areas and to set up constants without regard for their actual locations in memory. Through the use of assembly control statements, the programmer controls a wide variety of functions related to the process of creating an object program. Data processing statements are the symbolic instructions which are converted by the assembler into the machine-language commands in an object program.

Programming is greatly simplified through the use of macro instructions. These instructions, when specialized by the library processor, cause the insertion of library routines into a source program. One macro call to the library causes a whole routine to be inserted in a symbolic program.



Another means for facilitating programming is the use of literals. Binary, decimal, octal and alphanumeric literals enable the programmer to write in the operands field of a symbolic instruction the actual **data** (as opposed to the **address** of the field containing the data) to be operated on by the instruction. Area-defining and address literals may also be employed. An area-defining literal is used to define and reserve a working area in memory without using a separate data formatting statement. An address literal enables the programmer to specify a symbolic address in the operands field of an instruction in such a way that the assembler will use the address as an operand.

Both indexed and indirect addressing can be indicated in Easycoder symbolic language. Symbolic tags of from two to six characters can be used with both versions of Easycoder. In addition, Easycoder D allows the use of ten-character tags for increased flexibility.

LIBRARY PROCESSING

The library processors facilitate programming by allowing the programmer to utilize precoded program segments (macro routines) in his program. These macro routines, which are stored in a library file for easy reference, consist of frequently used sequences of instructions in a generalized form. The library processors accept macro calls written in a source program, obtain the macro routines called, specialize them according to parameters submitted by the programmer, and insert them into an

(Continued on reverse side)

Specifications remain subject to change in order to allow the introduction of design improvements.

122.1105.001C.0-437
8666
Printed in U.S.A.

When ordering this publication please specify Title and Underscored portion of File Number.

Honeywell
ELECTRONIC DATA PROCESSING

Easycoder symbolic program. The library processors can be used to specialize Honeywell-supplied library routines and/or generalized routines written by the user. All levels of nested macro routines can be specialized.

Another function of the library processors is the respecialization of macro routines in an assembled program on a symbolic program tape file. On a subsequent assembly run, the assembler replaces the old macro coding with the respecialized routines generated by the library processors.

A third function of the library processors is the punching of symbolic decks containing entire routines on the library file.

Library Processor C is used prior to the assembly of a source program by Easycoder Assembler C; Library Processor D processes input to Easycoder Assembler D.

EASYCODER ASSEMBLERS

The user may operate the assembler in one of four modes, enabling him to tailor his machine operation to his exact programming needs.

1. **Assembly.** The assembly mode is used when a new symbolic program tape file is desired. Programs are translated from symbolic language to machine language and stored on a symbolic program tape (SPT) in both symbolic and binary form. An assembly listing and a directory listing are produced. The assembly listing shows the program in source and octal codes; each programming error is flagged and diagnosed. The directory listing shows the order of all programs on the SPT. Both listings can be recorded on tape for off-line printing if desired.

   The binary run tape (BRT), another output of the assembly mode, contains the assembled program in machine-language format only. Programs may be loaded from the BRT Tape Loader-Monitor C or Floating Tape Loader-Monitor C and executed directly, or this tape may serve as the input to Update and Select C or D.

   Optionally the assembled program may be punched on a binary run deck (BRD) and loaded for execution by Card Loader-Monitor B.

2. **Selection.** The selection mode makes it possible to select previously assembled programs from an SPT file and place them on a BRT or BRD in preparation for execution. The selection process is controlled by a director deck.

3. **Assembly and Updating.** The assembly and updating mode is used to keep the master SPT current. In this mode, several operations can be performed on the master SPT file. New programs can be assembled and added to the SPT; obsolete programs can be deleted from the file; and corrections can be made to individual programs on the tape. Assembly and updating operation is controlled by a director deck. Besides the updated SPT file, assembly and directory listings are produced in this mode. Optionally, a BRT or BRD is also created.

4. **Assembly, Updating, and Selection.** This mode combines the functions of the first three modes in a single run, allowing the user to assemble and add new programs to the master SPT file, delete obsolete programs, form the tape and make corrections to individual programs in the file. In addition, specified programs are selected from the file and placed on a BRT or BRD for execution. Both assembly and directory listings are produced in this mode.

ADDITIONAL ROUTINES

The following utility routines, supplied by Honeywell, enhance the capabilities of Easycoder.

1. Analyzer C produces a handy cross-reference listing of all symbolic tags, macro calls, references to index registers, and references to absolute addresses used in a symbolic program.

2. BRT Punch C, a routine which converts BRT object programs to cards punched in binary-run format, is particularly useful in configurations with a small number of tape drives.

3. Through the use of SPT Merge C, assembled programs from several SPT's can be consolidated on one master SPT file to permit faster and easier handling of programs.

4. The Update and Select programs, which are used to maintain tape files of binary executable programs and to select programs from these files for execution, enable the user to combine in a single run programs originally prepared in Easycoder, Fortran, and COBOL languages.

EQUIPMENT REQUIREMENTS

Easycoder Assembly operates in an equipment configuration which includes three tape drives, a card reader, a printer, and the advanced programming instructions. Memory requirements are 12,288 characters for Library Processor C and Easycoder Assembler C, and 16,384 characters for Library Processor D and Easycoder Assembler D. Up to four additional tape drives can be included to allow several operations during a single run. Paper tape equipment is optional.