This is an informal report intended primarily for internal or limited external distribution. (The opinions and conclusions stated are those of the author and may or may not be those of the laboratory.) This report is not to be given additional external distribution or cited in external documents without the consent of the author or LBL Technical Information Department.

LAWRENCE LIVERMORE LABORATORY
University of California/Livermore, California

PL/M--A HIGH LEVEL LANGUAGE
FOR THE INTEL MCS-8 8008 CPU

W. G. MAGNUSON JR.
T. G. ALLISON
8/14/73

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.


DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED
DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
SUMMARY

This report is a user's manual for the CDC-7600 computer program PL/M. A reference manual for the PL/M language "A Guide to PL/M Programming" (1) is available from the Manufacturing and Materials Group, Electronics Engineering Department.

PL/M is a high level language and is structurally similar to IBM's PL/1 (2). In particular it resembles the XPL language (3). From the high level language input PL/M checks for syntactical and system violations. In Pass One, In Pass Two the program generates 8008 'BNFF' data, a map of the original statements versus memory locations, a relocated symbol table address listing, and a list of the assembly language generated by the compiler. Pass two also does the actual code generation for the MCS-8 8008 CPU.

PL/M allows programming in a high level language rather than in assembly language. The programs produced with PL/M will execute on the MCS-8 CPU with little or no loss in execution efficiency when compared to assembly language programming. Likewise the programs should be easier to maintain as well as a great deal faster to develop.

Reading The PL/M Program From Photostore

The PL/M compiler program is stored in the ELEPHANT Photostore under the "Take" directory:

.55885phony:INTEL

To retrieve the PL/M program for use on any of the CDC-7600 computers type the following:

ELF / .5 .1
.RDS .55885phony:INTEL:PL/M
.END
.RDS
.ALL DONE

Running Pass-1 of PL/M From the TTY

To invoke Pass-1 of the PL/M compiler type the following:

X PL/M PLM1 / 1 .5
.TYPE INPUT FILE OR "TTY"
.PLIMTR2 (for example)
.Ø PROGRAM ERRORS
.ALL DONE
Running Pass-1 of PL/M From the TTY (continued)

When running from a disk file, the file must be in packed-ascii format. The input file name must be seven or fewer characters in length.

Pass-1 will generate three output files. Their names and contents are as follows:

- **PAS1OUT**: Lists all input, errors, and an un-relocated symbol table (see Figure 1). This file may be listed on a printer or viewed on TMDS.
- **PAS1COD**: Intermediate compiler code for use by Pass-2.
- **PAS1SYM**: Intermediate symbol table for use by Pass-2.

If you have zero errors from Pass-1, then you are ready to proceed with Pass 2. Otherwise, the errors should be corrected and Pass-1 run again.

During Pass-1 only the $ANALYZE... and the $GENERATE compiler switches may be used (see page 52 of the programming guide). The other compiler switches have either been preset or are not applicable to the way the program is run at LLL.

Running Pass-2 of PL/M

To invoke Pass-2 of PL/M, type the following:

```
X PL/M PIM2 / 1 .5
TYPE LINE-FEED OR TAPE VAULT NO.
@ PROGRAM ERRORS
ALL DONE
```

Pass-2 will generate three output files. Their names and contents are as follows:

- **PAS2OUT**: Gives a map of original statement numbers (card numbers) versus memory location; gives the relocated symbol table addresses; gives a list of the assembly language generated by the PL/M compiler (see Figure 2).
- **BNPF**: This is the "BNPF" data which may be punched on cards and then converted to paper tape for loading the MCS-8 (see Figure 3).
- **PAS2SYM**: This is the symbol table output for use, together with BNPF file, with the MCS8SIM program. (See Figure 3).
Running Pass-2 of PL/M (continued)

When a Tape Vault Number is input in a Pass-2 run, then the BNPF file will be written onto it for conversion to punch paper tape for use with the PROM programmer. The CDC-160A computer is used for punching a paper tape using the magnetic tape.

In Pass-2 only the `$ANALYZE`, `$GENERATE=2`, `$HEADER=n`, and `$VARIABLES=n` compiler switches are available to use.

Need Help?

In running the PL/M compiler program on the CDC-7600 system if you have problems with the language then usually the answer will be in the reference 1. If you wish to follow a compiler run with a simulation run then refer to LER72-103403. Terry Allison (Computations) and Waldo Magnuson also can help you if your run into problems running the program or in punching a paper tape on the CDC-160A.

References


W. G. Magnuson, Jr.
Electronics Engineering Department

T. G. Allison
Computations Department

dla

Distribution:

EE Engineers
W. G. Magnuson, Jr. (20 copies)
T. G. Allison
L. L. Cleland
H. C. McDonald
H. W. Van Ness
Patent Engineering
DECLARE PRIME(50) BYTE;
DECLARE (1,0) BYTE;
DECLARE TRUE LITERALLY '1', FALSE LITERALLY '0';

PRIME(1) = TRUE; /* 1 IS A PRIME */

DO I = 2 TO 50:
PRIME(I) = FALSE; /* INITIALIZE TABLE TO FALSE */
K = 2;
DO WHILE I MOD K <> 0: /* LOOP UNTIL TEST FOR PRIME Fails */
   K = K + 1;
END;
IF K = I THEN
   DO; /* FOUND A PRIME */
      PRIME(I) = TRUE;
   END;
END;
EDF /* END OF PROGRAM */

NO PROGRAM ERRORS

Figure 1
PASLOUT Listing
Figure 3

NO PROGRAM ERRORS

PAS2SYM and BNFF Listings
NOTICE

"This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately-owned rights."