

IMPLEMENTATION OF AN INTEL 8080 MICROPROCESSOR DEVELOPMENT SYSTEM USING EXISTING MINICOMPUTERS

P.J. Horne
CERN, Geneva, Switzerland

Summary

The INTEL 8080 has become one of the two most widely used microprocessors (μ P's) at CERN, the other being the MOTOROLA 6800. Even though this is the case, there has been up to date only rudimentary facilities available for aiding the development of application programs for this μ P. An ideal development system is one which has a sophisticated editing and filing capability, an assembler/compiler, and access to the μ P application. In many instances one also requires access to a PROM programmer as the application may utilize only PROMs for program storage. With these thoughts in mind, an INTEL 8080 μ P development system was implemented in the PS Division at CERN. While any other minicomputer would be suitable, this system utilizes a PDP-11/45 minicomputer as the editing and file-handling machine, and an MCS 8/ MOD 80 μ P development system for assembling, PROM programming, and debugging user programs at run time. The two computers are linked by a CAMAC crate system, which provides the means of access to μ P applications in CAMAC and the interface of the development system to other applications.

Introduction

Program development is the most time-consuming (and therefore expensive) section of the majority of microprocessor applications^{1,2}. To keep this time to a minimum, it is necessary to have as good a development system as possible to aid the user in implementing his application programs for a particular microprocessor (in this case the INTEL 8080). On the arrival of the INTEL 8080 on the market, the only development system available was the MCS 8/MOD 80 microprocessor development system. This microcomputer was acquired by the Booster Group for aiding the development of software for a particular application^{3,4}. However, it soon became apparent that if any extension of the application was required or any other application arose, a more sophisticated development system would be needed.

By program development is meant the following sequence of steps:

1. Specify the required actions in the form of a flow-chart or high-level language.
2. Translation of this chart or high-level language into the assembler language of the particular microprocessor.
3. "Edit" the program into a source file.
4. "Assemble" the source file to obtain the binary file.
5. "On-line debugging" of the program.

The time required to perform steps 1 and 2 is independent of the development system (assuming no high-level language compiler is available), while steps 3, 4, and 5 are system-dependent.

A good development system is one that gives good support to the user during steps 3, 4, and 5 of the program development, i.e. it has good file-handling, editing, assembling, and debugging facilities. The MCS system has a good assembler and debugging facility, but a poor editor and no file-handling (as it has no mass storage device such as a disk) except that of paper-tape reading and punching. There was, however, an extremely good development system for the PDP-11 minicomputers near at hand, which has sophisticated file-handling and

editing facilities. It was therefore decided to "link" the MCS to the PDP-11/45 to obtain a more sophisticated microprocessor development system (see Fig. 1). As the PDP-11/45 MCS was already connected to a CAMAC crate system, its interface to the MCS was relatively simple.

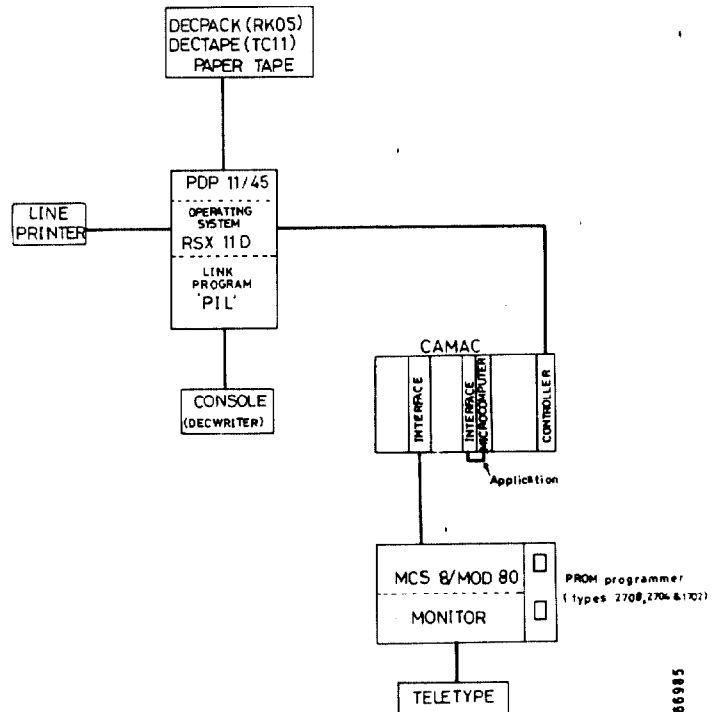


Fig. 1 INTEL 8080 development system

On the software front the incorporation of a high-level language, e.g. CORAL 66, BCPL, PL/M⁵⁻⁷, for microprocessors into the development system is part of the on-going implementation. An ideal language should be easy for the non-computer scientist/engineer to learn. It should aid him in writing structured programs and, of course, be available on a minicomputer. It should also be possible to include in the language algorithms that are specific to the particular applications domain specific features⁸. For example, for CAMAC applications the compiler must know the keywords LAM, CAMAC, C, N, A, F, etc., and generate the required object code.

Implementation

The MCS contains a built-in operating system MONITOR VERSION 3.0⁹ on EPROM, which may be started by a simple bootstrapping via its front panel. This monitor allows the user to access the memory via dump or display commands, control a user program and program EPROMs, type 1702A, 2704 & 2708 INTEL EPROMs. A list of all the available commands is given in Ref. 9. A detailed description of the system is given elsewhere¹⁰.

It was decided to utilize this monitor -- and not treat the MCS as a special memory extension to the PDP-11/45 -- in such a way that all data transfers between the two machines would be in the form of ASCII strings which the monitor should interpret before giving a response. Each byte transferred is echoed by both machines

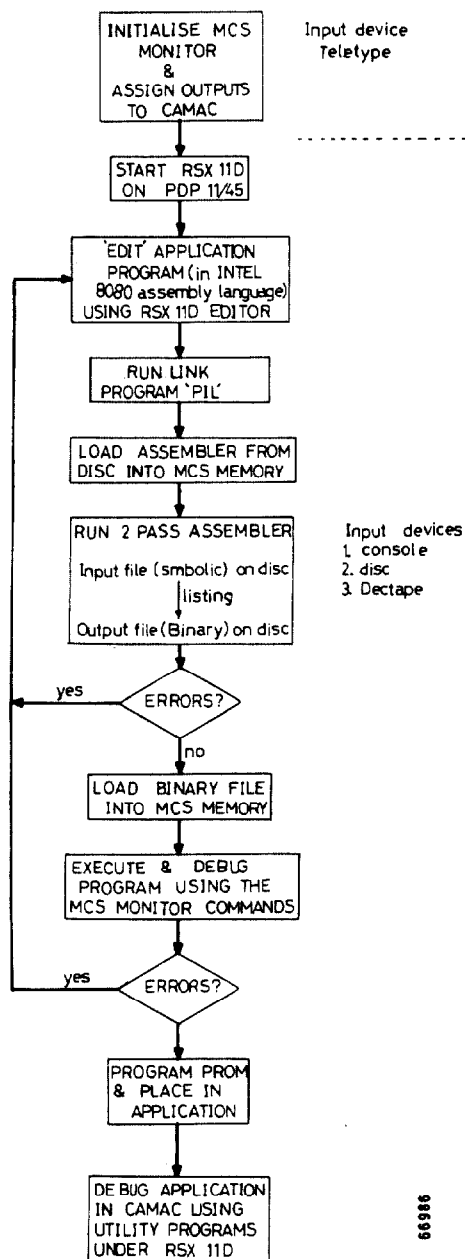


Fig. 2 Flow chart example of using the development system

making synchronization of the program 'PIL' to the MCS monitor easier and providing some basic error-checking facility.

The software to implement the link was written in PL-11¹¹⁾ and runs as a privileged task under the RSX 11D operating system on the PDP-11/45. PL-11 was chosen as the language for writing the link program because of its I/O support library and its speed of operation with the CAMAC system.

With this link implemented, program development for the INTEL 8080 μ P is improved (time-wise) by at least a factor of 50 and enables the user to utilize all the I/O devices of the PDP-11/45 (which are: two DEC'packs, two DEC tapes, a teletype, and a line printer) thus providing a powerful microprocessor development tool achieved by the use of existing equipment and a very simple interface module. The sequence of events required to develop a program using this development system is shown in Fig. 2.

References

- 1) Software becomes the challenge, *Electronics* **49**, No. 8, 104 (1976).
- 2) R.D. Catterton and G.S. Carilli, *Electronics* **49**, No. 19, 91 (1976).
- 3) G. Benincasa and B. Hallgren, Using a microprocessor in an improved varilog facility, CERN MPS/BR Note 75-2 (1975).
- 4) P.J. Horne, Implementation of a varilog for the PSB multipole power supplies using a microprocessor, CERN PS/BR Note 76-24 (1976).
- 5) P.M. Woodward, P.R. Wetherall and B. Gorman, Official definition of CORAL 66 (Royal Radar Establishment, Malvern, Worcs., 1976).
- 6) I. Willers and S. Meller, the BCPL mini-manual, CERN Data Handling Division, April 1976.
- 7) Intel Corporation, 8008 and 8080 PL/M programming manual (Intel Corp., Santa Clara, Calif., 1975).
- 8) S.A. Ward and P.G. Jerrel, Proc. 2nd Symposium on Microprocessors and Microprogramming, Venice, 1976 (North Holland, Amsterdam, 1976), p. 197.
- 9) Intel Corporation, INTELLEC 8/MOD 80 operators manual (Intel Corp., Santa Clara, Calif., 1974).
- 10) P.J. Horne, An INTEL 8080 microprocessor development system, CERN 77-02 (1977).
- 11) R.D. Russell, PL-11, A programming language for the DEC PDP-11 computer, CERN 74-24 (1974).