

MICROCOMPUTER SOFTWARE DEVELOPMENT FACILITIES

J. S. Gorman and C. Mathiasen
Ford Aerospace
Houston, Texas

Historically, microcomputer software/firmware has been developed on a system designed to perform software development and emulation functions for a specific processor. This method proved successful during the early years of microcomputer software development. However, both the number of microcomputers used in system design, and the complexity of the microcomputer software, have increased. These changes dictate that more efficient and cost effective methods be found for developing microcomputer software.

One approach is to utilize a host computer with high-speed peripheral support. Application programs such as cross assemblers, loaders, and simulators are implemented in the host computer for each of the microcomputers for which software development is a requirement. The host computer is configured to operate in a time-share mode for multi-users. The provided remote terminals, printers, and down loading capabilities are based on user requirements. With this configuration a user, either local or remote, can use the host computer for microcomputer software development. Once the software is developed (through the code and modular debug stage) it can be downloaded to the development system or emulator in a test area where hardware/software integration functions can proceed. The microcomputer software program sources reside in the host computer and can be edited, assembled, loaded, and then downloaded as required until the software development project has been completed.

The use of a host computer for microcomputer software development allows greater expansion and versatility and has resulted in increased performance and improved programmer productivity.

Preceding Page Blank

Microcomputer Software Development History

- Traditionally accomplished on dedicated development systems
- Development systems supported one specific processor or processor sets
- Minimal operating systems or monitor programs supplied for run time and development support
- Standard utilities provided: assemblers and editors
- Development method initially adequate

The Need for Change

- Increasing software complexity surfaced many problems
- Single user environment time consuming, inefficient
- Difficult to support more than one type of target processor
- Development systems provided minimum peripheral support
- Development capability relatively slow
- Non-availability of development system became serious problem
- Need for change eminent

Host Computer System Philosophy

- Use of host computer system solves many problems
- Host computer can service multi-user environment
- Via cross-product software, one host computer can support any number of target processors
- Host computer peripheral support can be wide and varied
- Host computer provides substantial, resident development tools
- Host computer can support remote development facilities
- Once software developed, can be downloaded to target processor
- Debug and integration can continue via in-circuit emulators

Implementation: Hardware Implications

- Terminal, printer, and download line selection can be determined from user requirements
- Communication equipment selection can be determined by data rates, line protocol, transmission distances, and signal multiplexing
- Computer interfacing equipment selection can be determined by I/O availability and user equipment interface characteristics
- Remote printers and host computers must support busy/ready indication using modem circuits
- Using standard host computer interface protocol in user equipment can minimize computer interface driver modifications

Implementation: Software Implications

- First priority: selection of cross-product software
- If local talent and experience available, software products can be developed instead of purchased
- Off-the-shelf assemblers and high level language compilers will support most processors
- Most vendors supply executable format software only. Some vendors provide source code
- Software for simulation capability should also be purchased
- Via total cross-product software package and host system development facilities, software can be checked out through software-to-software integration phase

Possible Obstacle: Transferring Software to Target Processor

- Use like peripherals on both host system and target processor
- Host computer flexible disk system can simplify software transfer
- Host computer flexible disk system can also provide source code archival
- Flexible disk formats must be hardware and software compatible with target processor format
- If host computer is remote from target processor, shuttling diskettes back and forth can lead to additional software transfer method analysis

Developing A Downloader Capability

- Some vendors supply downloader software
- Development involves minimal software effort
- Downloader approach involves use of development system or emulation device
- Can use dedicated development system or emulator port for host computer interface
- Via download lines, can use I/O device on development system or emulator to communicate with host computer
- Two downloading philosophies
 - Controlled method
 - Pseudo terminal method
- Same environment will also support upload capability

Avoiding Pitfalls

- Consider many issues during planning stages
- Host computer memory limitations can also limit performance and efficiency
- Host computer task size limits can increase overlay needs
- Cross-product software idiosyncracies can limit software transportability
- Investigate symbol table and macro limitations before purchasing cross-product software
- If source code purchased, additional cost factors can surface
- Development facility personnel must be adequately trained in host computer use and cross-product software

Typical Microcomputer Software Development Facility Configuration

