machine-dependent parts of the language, the editor should flag any usage errors.

Simulators. A high-level simulator will interpret HLML statements and let the user trace a microprogram in terms of the HLML. A low-level simulator could simulate each microinstruction produced by the compiler and report on the effects of the microcode execution in both high-level and low-level terms.

This low-level simulator is necessary to debug a compiler for a new processor and to help the microprogrammer at the hardware interface. It is necessary when debugging the hardware of a new processor to have complete information about the effects of the microprogram at a low level. The simulator will provide this information.

The low-level simulator must be table-driven to simplify reconfiguring the simulator (and thus the programming environment) for other targets. Work in this area includes the E-Machine workbench[13] and research on using models of micro-operation format and semantics in simulator generation.[14]

In addition to a resource checking capability, these simulators will provide execution profiles and timing information. These are a great help in determining where the microprogram spends its time and are far easier to implement in a simulator than in a compiler.

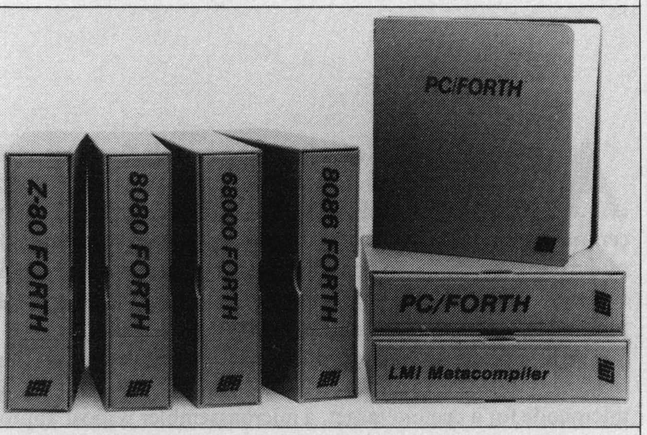
Assertions should also be a feature of the language and simulator. Instead of being executable (the usual case), these assertions should compile into breakpoint specifications for the simulator. If this is done, the real microprogram—not one altered by the presence of code for the assertions—can be simulated.

Performance requirements specification. Another part of the environment, which might be migrated into the language itself, should be a way of specifying the microprogram's response to asynchronous events. The specification must also be able to specify microprogram timing in the sense that a certain microcode segment must execute within a certain amount of time. Specifying this information in an understandable way is a difficult problem. Going from such a specification to code that satisfies requirements is a still more difficult problem. Certainly, research in real-time languages and systems will be valuable for microprogramming in the future.

Verification. A verification system might be included for some applications, but in general the cost of such a system—at least in the near future—would be excessive. This system would prove the correctness of the microprogram with respect to the given specifications. It will probably be interactive and will communicate with the microprogrammer through an interface similar to those used by the simulators.

A great deal of work has already been done on verification of microcode. This works suggests that a verification system, though expensive now, will become feasible for critical microprograms. An aid to making verification feasible is axiomatization of the microprogramming language. Work has been done in this area for S*(QM-1).[15]

TOTAL CONTROL
with *LMI FORTH*™



For Programming Professionals:

an expanding family of compatible, high-performance, Forth-83 Standard compilers for microcomputers

For Development:
Interactive Forth-83 Interpreter/Compilers
- 16-bit and 32-bit implementations
- Full screen editor and assembler
- Uses standard operating system files
- 400 page manual written in plain English
- Options include software floating point, arithmetic coprocessor support, symbolic debugger, native code compilers, and graphics support

For Applications: Forth-83 Metacompiler
- Unique table-driven multi-pass Forth compiler
- Compiles compact ROMable or disk-based applications
- Excellent error handling
- Produces headerless code, compiles from intermediate states, and performs conditional compilation
- Cross-compiles to 8080, Z-80, 8086, 68000, 6502, 8051, 8096, 1802, and 6303
- No license fee or royalty for compiled applications

For Speed: CForth Application Compiler
- Translates "high-level" Forth into in-line, optimized machine code
- Can generate ROMable code

Support Services for registered users:
- Technical Assistance Hotline
- Periodic newsletters and low-cost updates
- Bulletin Board System

Call or write for detailed product information and prices. Consulting and Educational Services available by special arrangement.

LMI ***Laboratory Microsystems Incorporated***
Post Office Box 10430, Marina del Rey, CA 90295
Phone credit card orders to: ***(213) 306-7412***

Overseas Distributors.
Germany: Forth-Systeme Angelika Flesch, Titisee-Neustadt, 7651-1665
UK: System Science Ltd., London, 01-248 0962
France: Micro-Sigma S.A.R.L., Paris, (1) 42.65.95.16
Japan: Southern Pacific Ltd., Yokohama, 045-314-9514
Australia: Wave-onic Associates, Wilson, W.A., (09) 451-2946

Reader Service Number 2