Microprocessor standards

by TOM PITTMAN
Itty Bitty Computers and Homebrew Computer Club
San Jose, California

and

ROBERT G. STEWART
Stewart Research Enterprises
Los Altos, California

Now that microprocessors have been with us for three or four years, we can stand back and observe some disadvantages of our free enterprise system, as well as obviously great advantages. The disadvantages tend to fall most heavily on the shoulders of the user, not the manufacturer. The proliferation of microprocessors with incompatible instruction sets, adds unnecessary chores to learning new mnemonics, and requires rewriting (at large expense) existing software. Another interesting situation can be seen in the case of the 8080 and the Z80 processors whose mnemonics differ for precisely the same instructions. The act of copyrighting mnemonics by microprocessor manufacturers places the problems in the user’s laps in order to protect the proprietary interests of the manufacturer of the chip.

Whenever standards are proposed, two reactions are voiced, polarizing the opinions. On the one hand we hear general disdain: “Who needs them?” or “Standards impede progress.” On the other hand we hear cheers: “What took so long?” or “Lack of standards impedes progress.” The field of microprocessors is not immune to this division.

We will have standards in software as well as in hardware. Unlike the larger mainframe computers, where the entry costs are high enough to insure a virtual monopoly on software and interface standards by a single manufacturer, there are a half dozen or more significant contributors to microprocessor software for each chip type, and as yet none of their products are compatible with each other. There are hundreds of components designed for very few processors, and many of these cannot even coexist in the same system. Standards are inevitable because the users will not tolerate this chaos very much longer.

The absence of standards for the large, batch computers today is simply due to the fact that the manufacturers fight standardization. Plus compatible peripherals are a touchy example. It is not in the mainframe companies interest to foster a standard interface. This reluctance to standardize extends to software aspects which directly affect a user such as job control streams, editor commands, disk I/O calls, tape formats, internal data representation, ASCII vs. EBCDIC, and pairing of source and relocatable files. These differences are particularly vexing for someone who must use computer systems made by a number of different companies. If there were only one computer company, then standardization efforts would not be necessary. The interesting thing is that the present situation looks like it results from a condition where all the different computer companies believe that they are the only one! Note that most of the problem areas which would benefit from standardization would have relatively little or no impact on the computer architecture!

The fun aspects of the hobby or homebrew computer run headlong into the software and hardware idiosyncrasies of the different manufacturers. For example, because the Port designations for the console and the console status flag’s ready bits have not been standardized, the act of bringing
up a system is rendered much harder since a friend with a working system may not be able to substitute his proven software without first patching in the required changes. This patching can be quite difficult because the hardware may be flaky also.

Note that these types of problems are seldom serious to the manufacturer, but represent hurdles to the novice user that make him curse the people responsible for making life difficult for him. And typically those people are cloistered within companies and make arbitrary decisions without regard to the problems that result for users interfacing with hardware and software from a number of sources.

A veritable jungle has come into existence in the home-brew or personal computer hobby. Memory boards and DMA devices for the S-100 bus are an example. The manufacturers state that they are S-100 compatible, and indeed they are unto themselves. However memory boards from different manufacturers are, in some cases, not compatible with each other. The initial MITS description of the Altair or S-100 Bus did not nail down the timing of the bus. By this absence, the timing defined by Intel for the 8080 processor must intrinsically be recognized as the original timing standard for that bus. Later, other processors such as the Z-80 and the 2650, have been used on the bus, without having the same timing signals and relationship as the 8080, so that memory and DMA devices are not directly interchangeable.

Usually it is the user who takes the lumps in both cost and frustration because the manufacturer had stated the boards are bus compatible, when they actually were not because of their timing and control characteristics.

The IEEE Computer Society Microprocessor Standards Committee was formed in the summer of 1977 to try and resolve some of the obvious problems that have arisen in the use and applications of these marvelous chips called microprocessors. The veritable complete revolution in technique which the microprocessor has brought to the electrical engineering profession will be aided for years to come by a codification attempt at this time.

SOFTWARE STANDARDIZATION EFFORTS

The IEEE Computer Society has taken a bold step forward in forming a standards committee to deal with the most pressing of these issues. We have selected three specific areas of concern in the software area where immediate action can be of tremendous benefit in the microprocessor community:

1. Floating point format and algorithms—We expect to be soliciting comments on a first draft of a proposed standard by the time of NCC. In this standard we will be defining forms for both normal floating point operations and for extended range and/or precision. Overflow and underflow conditions will be specified, and minimal requirements of the algorithms implementing these operations will be defined.

2. Assembly language syntax and lexicography—There are major questions subject to compromise in this area. It is expected that this standard will define the assembler syntax for each of the more popular microprocessors, specifying both mnemonics and operand coding. By considering all of the processors at one time it is hoped that the similarities between processors will be reflected in the mnemonics and operand syntax, and that guidelines may be developed for the assignment of mnemonics to new microprocessors as they appear. In particular, it is hoped that assemblers supported by timesharing services will all accept the same source code for the same chips (this is not now the case), and that their reduced development costs in bringing up additional processors will make wider support for new CPUs more attractive.

3. Relocatable object module formats—This area, though of vital importance, seems to be moving the slowest. Again, the major thrust of our efforts is the specification of an adequate object module format for each of the major processor families.

Here also it is expected that the careful selection of functional and lexical characteristics will insure a uniformity across processor lines which will therefore also extend into future CPU design. Clearly the burdens placed on the relocating linker are somewhat arbitrary, so part of our work includes the analysis of the kinds and extent of the tasks which should be deferred to that phase of program translation. Obvious candidates for support are the output of macro assemblers (see standards effort #2 above) and of FORTRAN compilers. The requirements of other high level language compilers should also be considered, as well as the possibility of a proper subset (with reduced functions) for minimal operating systems.
One of the frequent objections we hear points to the dissimilarities between different microprocessors, with particular reference to the one-chip controllers. Since most of these are single-sourced and lack the multiple-manufacturer proliferation of software development tools seen in the 8-bit processors like the 8080 and the 6800, we have not been overly concerned about extending compatibility in these directions. However, most of the committee members have had extensive experience with several microprocessors, and our emphasis in each of three areas is focused on the similarities rather than the differences between the varieties of chips under consideration.

**HARDWARE STANDARDIZATION EFFORTS**

The initial hardware problems tackled by the Microprocessor Standards Committee are standardization of several of the existing and proposed busses, namely the S-100 bus, the Intel MDS Bus, National's Microbus, and the PI bus proposed by Pietsch and Khalsa. After the bus issues have crystallized, the Committee will look at problems in the Small Business and Hobby Computer area, such as interfacing and protocols, standards for memory peripherals, data formats, disk formats, etc.

The key issues that arise in standardizing the de facto busses are timing diagrams, DMA control, defining unused pins for future address, data space, and control extensions, and accommodating the higher clock rates of future processors. The proposed standards for the busses will be given at NCC and the audience will have the opportunity to make comments on them. They will also be able to make suggestions for future standards activity by the committee.

Of the existing de facto busses for microprocessors, the most widely used is the bus MITS used on its Altair 8800 computer announced by Ed Roberts in Popular Electronics magazine in February, 1975. (Many computer hobbyists regard that as the world's first digital computer.) The wide variety of plug-in boards that are available for the Altair of S-100, at competitive prices, have enabled the bus to serve the hobbyist and the small business market reasonably satisfactorily. The problems that have arisen with the S-100 bus are: (1) The use of positive true rather than negative true logic, (2) an overabundance of control and non-essential signals on the bus, (3) the use of separate data in and data out busses, (4) the use of microprocessors with different timing characteristics than the 8080, (5) different voltage power lines directly adjacent to each other, (6) one common AC and DC ground, and to some extent, (7) poorly designed boards which plug into the bus.

The efforts of the IEEE Computer Society Microprocessor Standards Committee to resolve some of the S-100 bus problems have been spearheaded by Howard Fullmer and George Morrow. The timing and DMA protocols adopted define processor-memory interactions in a manner which is more tolerant of non-8080 processors. All timing relationships are specified only with respect to the phase two (d2) clock signal. Any device proclaiming itself to be a bus master, such as a DMA device as well as a processor must generate a set of 18 control signals as well as the address and data signals. Some DMA devices have failed to generate the full set of bus control signals and so have led to incompatibility problems with different memory boards which were relying on the presence of those signals. The 8224 system controller can be used by DMA devices and non-8080 processors to generate the control signals needed on the S-100 Bus. The adoption of standards for the bus will enhance its utility in the years to come and help preserve the investment which tens of thousands of hobbyists have made in their equipment.

The capability of the chip manufacturers to develop ROM chips to store specialized software also will act to force standards on higher level languages as they relate to microprocessors, development systems, operating systems, and the personal computing market. Such chips will provide BASIC, FORTRAN, PASCAL etc. interpreters and compilers at the selection of the user. Thus the advance in hardware technology acts to force upon us the standardization of relocatable code formats discussed earlier.

Packaging standardization for microprocessors has evolved to the predominant use of a 40 pin package. As longer word length processors are created, more pins may well be required unless extensive bus multiplexing is used. There is some tendency now to use pins on opposite diagonals for ground (pin 20) and power supply (pin 40); however, even that very limited pin-out standardization is not at all universal. Chip designers lay out the chip with all the logic capability they can, and only worry about the package pin-out at the end. If a pin-out standard is imposed for microprocessors, then the chip designer would have to consider the pin-out in the initial chip design, and may have to run traces for longer lengths than otherwise in order to reach the pads. This acts to reduce the available chip area for the device's logic functions. Thus there are significant semiconductor chip layout and utilization reasons to deter the formulation of a pin-out standard of μP packages.

A far more significant aspect of microprocessor architecture and design which would benefit from standardization right now is the "timing architecture," or precise sequence of signals that interface with external busses and support chips. In the practical sense of being able to use existing support chips and busses for a new or different microprocessor, the timing architecture is as (or more) important a conceptual issue as the register structure and logic capability. The dominant existing de facto busses have intrinsically been designed around the timing convention of the 8080. A timing standard for handshaking and control signals of microprocessors would be very useful in furthering the universality of busses, support chips, DMA devices, and controllers. Such a standard should have to be processor independent if at all possible. The work which our committee has undertaken on the S-100 bus timing is essentially committed to the same goals, and will probably be helpful in developing a standard timing architecture for microprocessors themselves. The committee has attempted to choose specific tasks which do not impact architecture in the usual sense, and the microprocessor manufacturers in Silicon Valley agree that our tasks would not.
The efforts by U.S. governmental agencies to impose standards on computers has resulted in the selection of a subset of existing architectures as being acceptable for future DOD procurement. Further, the instruction set of the AN/UYK 20 minicomputer is being established as the standard required instruction set for all future minicomputers. In addition, some governmental agency staffs active in standardization efforts believe that a standard bit pattern for the processor word is achievable for a given instruction. If such a requirement is imposed in the future, then the specific processor design and architecture will certainly be impacted. An instruction set standard can be dealt with by an assembler or cross-assembler; imposing a standard bit pattern in the word would require hardware or a chip design that would implement the standard. Whether the conventions that have been established by DOD for computers and minicomputers are forced downward on microprocessors remains to be seen.

The rapid changes in technology have led to multiple sources of software and hardware to create a system, as compared to a single source, bundled situation which has prevailed in the past. For the user to benefit from the remarkable advances in technological capability, it really is essential to formulate suitable software and hardware standards now.

ACKNOWLEDGMENTS

The cartoons by "Sandy" are copyrighted by Creative Computing Press and are used with permission. The authors have benefited greatly from conversations with other committee members, and the ideas and attitudes expressed herein are shared by many of them.