De facto standards will not be adequate to allow widespread communication among personal computing systems. Here is a look at the issues and the possibilities.

MICROSYSTEMS

Standards for the Personal Computing Network

James Isaak
Data General

In 1972, the technology for personal information processing took an imperceptible but critical step. Before that time, a timesharing system based on cable television was the most likely way to get computing benefits to the masses,[1] but in 1972 the microcomputer entered the market and opened the door for personal computers. Many of the high impact applications—electronic mail, remote shopping, financial transactions, and education—necessitate standards for communication between systems.[2] Even without a telephone or CATV connection, the potential for a network exists, and standards for information transfer must be formulated.[3,4]

Why standards?

The personal computing market is developing from three high-growth-rate product areas—hobby computers, calculators, and video games.[5,6] It is reasonable to assume that manufacturers entering the market will try to differentiate their products (and ensure growth) by keeping them incompatible. If so, the consumer—and quite possibly our national communication/information potential as well—will suffer, since differentiation will limit both consumer options and utility value of a communication/information network.

Standards could be established before this market gets locked into either too limited or incompatible equipment. A personal computing network could move us from the industrial age to the "age of information," with all of the impact and benefits that implies. We are on the threshold of a second Babel, and the resulting confusion of communication could be as counter-productive as was the first Babel.[6]

The forces working against standardization are many. What is the motive for manufacturers' setting and adhering to standards? There is none; in fact it works in reverse. A manufacturer who isn't primarily seeking profit will not be around long enough to set a standard. It is most profitable for a corporation to set its own standard, then price its entry level equipment cheap and clean up on the additions. The generality of computer systems, the eccentricities of designers and programmers, and Murphy's law all conspire with the manufacturer's profit motive to ensure that no standard prevails. Progress itself is a co-conspirator. Just as most of the communication and transportation technology developed between 1880 and 1920 is now obsolete, so will the technologies of today be replaced tomorrow.[7,8] If we can keep the basic systems modular and expandable, however, the benefits of progress can be easily integrated.

Where do standards apply?

Because their target market is unsophisticated end users, these systems must be modular, self-diagnosing, and easily expanded or modified. The main hardware element is a microprocessor with some central memory. Likely components include high-speed rotating memory using CCD or bubble technology, a TV interface, a keyboard, a joystick(s), a communications controller and modem, and I/O media such as cassettes or floppy disks.[3,8] Many expansion devices beyond these are possible—and likely.

As an interface to the outside world, data communication is the most obvious area for standards. However, media and encoding formats are equally important for information exchange and distribution of programs on a mass basis. The interface to the wide variety of potential peripherals and control devices is a third area for standards. The motive for

0018-9162/78/1000-0060$00.75 © 1978 IEEE

peripheral interface standards is encouragement of a wide variety of interfaces and therefore applications by providing a large market. Finally, if these systems are going to transfer programs, then standard languages will be needed. Memory bus standards are probably useless since the central memory will soon be on the same chip as the processor.[8]

Communications standards. Data communication over the phone system represents one of the most significant capabilities of the home computer. With the potential for computer conferencing, electronic funds transfer, mail, education and many other services, its impact on our life styles could match that of TV, radio, and telephone. As with those communications links, it is the most obvious place for standards and for control by agencies such as the FCC.

The existing standard in hobby computers is a simple asynchronous, 300-baud (or 110-baud) phone link.[9] It is inexpensive and represents a common interface for local keyboards, terminals, and cassette recorders, as well as for telecommunications. However, as the standard for the home market, this technology has some major drawbacks. The method is slow and has no standard for error detection or correction. Moreover, since connection ties up a full phone circuit well below capacity, widespread use could substantially impact the phone system.

Fortunately, another standard developing in the industry—ADCCP— could provide significant relief for these problems.[10-14] Advanced Data Communications Control Procedure has been accepted as a standard in the US (ANSI-ADDCP) and internationally (ISO-HDLC and CCITT X.25). Most significantly, it has been accepted by communications companies such as Telenet and Bell Canada. For the home computer industry, key elements of the ADCCP standard are its error handling capability and its potential for multiplexing messages on a single communication line (via packet switching). Systems designed with proper interface hardware and software (or firmware) could connect directly into a packet switching network. Even the cost should drop significantly as mass production for industry and communication carrier support increases. Alternatives to the phone system, such as CATV or radio connections, could be used easily with this protocol as well.[15]

It is possible to establish nodes which allow interfacing older asynchronous systems to a packet network, but this will place some serious limits on their possible services. In general, only one unit could run off a single phone connection, and services requiring significant amounts of data (news services or library access) would not be practical.

Media and encoding formats. Exchange of software and data and use of long-term storage both require machine-readable hard copy. The commercial world depends on cards, paper tape, or magnetic tape for this purpose, yet there are very few recognized standards. For cards, "029 80 column" probably comes the closest; for magnetic tape, "NRZI, 800 bpi, 9 track, unlabeled, one 80-character record per block, EBCDIC" reflects the kind of specification required to read a tape. The actual media, hardware structure, and encoding techniques differ radically.

The hobbyist market started with paper tape but without low-cost punch units. There is strong pressure to convert to a cassette-tape standard medium. The basic concept is to hook the recorder into the modem-style interface, directly copy the acoustical signals, and then receive them back. The Kansas City standard translates this into specific frequencies, timings, and start-stop sequences.[16] Unfotunately, this is limited to the transfer of data without a concept of files—and at 300 baud. So discussion continues on how to separate files and improve speed.[17]

The cassette remains the most attractive medium for data distribution, with floppy disks a possibility.

The asynchronous communications problems listed above apply to this type of standard. Therefore, it seems proper to apply the same solution—ADCCP. By implementing cassette (and possibly diskette) recording with the bit-synchronous-oriented serializing of data, common software and hardware can be used to control communications and data storage. The control fields can be used to identify continuation records from file headers and provide control information to the interface. (Now, just stopping the tape after a successful read can be a problem, and rewind is virtually unheard of.) A significant spin-off from this technique is the distribution of software through phonograph records, which, like audio cassettes, can provide acoustical encoding. The low-cost availability of audio cassette units and phonographs is a strong plus for these media and encoding techniques.

A final technique for distribution, printed software, is worth mentioning. Data can be encoded and distributed in bar code like that used in the supermarket industry,[18] but this requires the addition of a light wand for reading in data and increases the difficulty of output for most hobbyist systems. The cassette remains the most attractive interchange medium, with inexpensive floppy disks a possibility for the future.

Bus standards. Within the hobbyist market, the bus protocol is certainly the most discussed standard, reflecting both the hardware orientation of current hobby systems and the economics of the competitive interface market. The most obvious de facto standard is the S-100 bus, first implemented by MITS on the Altair system. This bus has been proposed as a standard and used by other companies.[19] However, significant issues have been raised about its design characteristics.[3,20] Even the "standardization" of the S-100 bus is not clear, since interfaces claiming to be S-100 compatible do not function in all implementations claiming S-100 buses.[21,22]

A second standard is the IEEE 488 bus designed for instrumentation and control. Currently used in the hobby market only by the Commodore PET,[23] it applies asynchronous communication as the bus protocol. Again, this is a step towards commonality with the telecommunications interface. It also separates the memory and I/O buses, an obvious step as more memory is built into the microprocessor chip or CPU card.

The current "high-level" hobbyist language is Basic—tiny Basic, this Basic, that Basic—but not a standard Basic.

For terminal devices, cassettes, reader/punches, and other low-speed peripherals, the simplest standard is that used for the telecommunication link. Again, it would be useful to standardize on the ADCCP protocol, allowing common software and hardware support. Devices must be "smarter" to provide this interface; however, a single controller could drive many such devices, offsetting the space and power limitations of card chassis and power supplies. High-speed peripherals would probably require a separate bus (communication port) to support high transfer rates and direct memory access (if it is not supported by all the ADCCP interface ports on a system), but these could still use the ADCCP protocol.

Language standards. Here we find (as has been the history of computers) Babel in full force. Even when the languages currently in use have the same name, they rarely have the same structure or meet standards of any kind.[3,21] Since languages are almost totally software, the variations are large, and the incentive for differentation by manufacturers is great. To get agreement on standards may actually require an act of Congress (see DOD-1 references[24,25]). But, if we are going to build a tower, we must put aside the throwing of stones.

The current "high-level" hobbyist language is certainly basic—tiny Basic, "this" Basic, "that" Basic—but not a standard Basic. Even minimal personal computer standards should include Basic, and help may be on the way with the eventual creation of an ANS Standard for Minimal Basic (x3J2/77).[26] Certainly Basic satisfies many objectives—simplicity, widespread implementation, and the availability of programmers trained in its use. However, Basic has a number of holes for general-purpose computing, in particular its lack of block and data structures.

A second area, the assembly language level,[21] appears as lacking in standards as Basic—if not worse. Although Intel and Zilog have chosen higher-level PL/I- and Pascal-styled implementations for system languages,[27,28] at this point they are not targeted at standards. Nor have they gained wide usage beyond the products of the companies themselves.[29]

A variation of the Pascal approach may be developing as an effective backup. First, the DOD language competition, clearly targeted on a Pascal-based standard, will result in significant pressure for vendor development of the selected standard.[24,29] Moreover, a second effort specifically aimed at the mini/micro world is taking place at the University of California at San Diego.[30] These efforts and moderate interest in more structured languages from both university and industry, may combine to make Pascal the ideal language for standardization. Pascal's few drawbacks in its initial definition[31] are worth overcoming in a standard. Key objections are the lack of strings as primitives (with associated functions) and the weak definition of I/O functions.[30] Other concerns are its lack of dynamic arrays and error or exception structures, and its limited redefinition of data storage areas.

The UCSD implementation is both an operating system and a language. Once these two have been established, the utilities, including support for Basic and other languages, are possible. The real key is establishing a common base in terms of system support and an efficient, powerful system language that can be coded in software, firmware, or even the chip design. The rest will follow, since other standards, like Basic, can be implemented just once. (Can you imagine not having to reinvent the wheel?) If all personal computing systems have the same basic foundation, it can be built upon without transportability, conversion, and similar problems that abound in the industry.

Notice that the level of specification is dictated by the desire to have the maximum communication compatibility between personal computing systems. These key communication elements are a common base language with full power potential, a common I/O standard at the signal level and from the programming viewpoint, a common protocol for information interchange in both hard copy (cassette/floppy disk) recording and file formatting, and telecommunications standards.

Standards implementation and enforcement

Establishment and enforcement of standards could fall in the government's domain (FCC-interstate communications standards) or be a fallout of some form of market dominance. Another alternative is a conscious effort to set standards, provide a basis for measurement, and evaluate systems against the standards. A combined effort of ACM, IEEE, computer hobbyist groups, and perhaps consumer product evaluation groups could yield such results.

Once agreement on areas where standards are necessary is reached, the concept of levels of implementation that are applied to Cobol could be applied to personal computing. These would cover hardware interfaces as well as software and system functions. In the language area, a level "0" implementation would be the essential subset needed to implement higher-level extensions. In this manner, higher levels could be implemented in hardware, in firmware, as an interpreter, or in a macro fashion. Finally a set of validation tests could be established as a basis for measurement. Each system could then be tested and described in terms of its level of compliance with or deviation from the standard. Consumer groups could provide comparisons and subjective comments as the ultimate guide. With well defined standards and published evaluations, a manufacturer would be risking a bad press if he deviated too far without substantial reason.

A final consideration is the ongoing need to review these standards. As we become more experienced and the technology changes, so must the standards. With some foresight we can establish an ongoing review process as well as the initial standards. Not since the telephone has such a potential for communication been possible. If we fail to act now to define standards, the result will surely be Babel. ■

References

1. E. B. Parker and D. A. Dunn, "Information Technology: Its Social Potential," *Science*, Vol. 176, June 1972, pp. 1392-1399.
2. A. Kay, "Micro Electronics and the Personal Computer," *Scientific American*, Vol. 237, No. 3, Sept. 1977, pp. 231-244.
3. J. Warren, "Personal and Hobby Computing: An Overview," *Computer*, Vol. 10, No. 5, Mar. 1977, pp. 10-21.
4. P. Isaacson, "Personal Computing Dissected," *Computer*, Vol. 10, No. 7, July 1977, pp. 71-73.
5. N. Lindgren, "Semiconductors Face the 80's," *IEEE Spectrum*, Vol. 14, No. 10, Oct. 1977, pp. 42-48.
6. C. Bradley, "Byte Industries' Nels Winkless Tells What's Happening," *People's Computers*, Vol. 6, No. 5, Mar. 1978, pp. 28-31.
7. R. N. Noyce, "Micro Electronics," *Scientific American*, Vol. 237, No. 3, Sept. 1977, pp. 62-69.
8. M. Shepherd, "Distributed Computer Power: A Key to Productivity," *Computer*, Vol. 10, No. 11, Nov. 1977, pp. 66-74.
9. M. Wilber, "CIE Net: A Design for a Network of Community Information Exchange," *Byte*, Feb., Mar., and Apr. 1978.
10. R. B. Hovey, "Packet Switching Networks Agree on Standards Interface," *Data Communications*, May/-June 1976, pp. 25-39.
11. M. Edwards, "Transaction Processing Shapes Hardware/Software Evolution," *Data Comm. User*, Oct. 1976, pp. 17-29.
12. R. Berglund, "Comparing Network Architectures," *Datamation*, Vol. 24, No. 2, Feb. 1978, pp. 79-85.
13. IBM Synchronous Data Link Control, General Information, IBM Corp., GA27-3093, File No. GENL-09, 1974.
14. *ANS Document X534/589 (ADCCP)*, available from CBEMA, 1828 L. Street N.W., Washington, DC 20036.
15. G. Falk and T. McQuillan, "Alternatives for Data Network Architectures," *Computer*, Vol. 10, No. 11, Nov. 1977, pp. 22-29.
16. "Byte-Kansas City Standard for Cassette Tape Recording," *Byte*, No. 7, Mar. 1976.
17. A.H. McDonough and M.P. Hammontre, "Cassette I/O Format—Standards Are Still Needed!" *Kilobaud*, No. 8, Aug. 1977, pp. 18-29.
18. W. Banks and R. Sanderson, "Samples of Machine Readable Printed Software," *Byte*, No. 16,Dec. 1976, pp. 12-17.
19. D. Denney and J. Broom, "Proposed Hobbyist-Standard Bus Structure" (S-100); *Dr. Dobb's Journal*, Aug. 1976, pp. 8-9.
20. L. Frenzel, letter defending Heathkit's not adopting the S-100 structure, *Dr. Dobb's Journal*, Oct. 1977, pp. 10-12.
21. R. Stewart, "Standards for Microprocessors," *Computer*, Vol. 11, No. 3, Mar. 1978, pp. 65-66.
22. "Here Comes the PET," *Computer*, Vol. 10, No. 10, Oct. 1977, pp. 88-92.
23. G. Morrow and H. Fullmer," Proposed Standard for the S-100 Bus," *Computer*, Vol. 11, No. 5, May 1978, pp. 84-90.
24. D. Fisher, "DoD's Common Programming Language Effort," *Computer*, Vol. 11, No. 3, Mar. 1978, pp. 24-33.
25. J. Lee, "Considerations for Future Programming Language Standards Activities," *CACM*, Vol. 20, No. 11, Nov. 1977, pp. 788-794.
26. ANS Committee x3 J2/77 (Draft, proposed programming language Minimal Basic), May 19077, available from CBEMA, 1828 L. Street N.W., Washington, DC 20036.
27. W. Brown, "Microsystems: Modular Programming in PL/M *Computer*, Vol. 11, No. 3, Mar. 1978, pp. 40-46.
28. C. Bass, "Microsystems: A Family of Systems Programming Languages for Microprocessors," *Computer*, Vol. 11, No. 3, Mar. 1978, pp. 34-39.
29. W. Kejka, "Microcomputer Standardization Concepts," *Computer*, Vol. 10, No. 2, Feb. 1977, pp. 54-56.
30. K. Bowles, "UCSD PASCAL", *SIGMIN Newsletter*, Vol. 4, No. 1, Feb. 1978.
31. N. Wirth, "The Programming Language PASCAL," *Acta Informatica 1*, 1971, pp. 35-63.

Jim Isaak has just been promoted to senior marketing specialist for the Microproducts Marketing Group at Data General; before that he held the position of systems engineering district manager. His most recent responsibilities have involved technical support for computer systems sales and services. Prior to rejoining Data General, Issak worked at Intel, Calma Co., and IBM. He received the BS degree in computer studies and the MSEE degree in computer engineering from Stanford University. He is a member of IEEE, ACM, and the Computer Society.