The wipe-activated keyboard debate continues

Editor:

Professor Montgomery's article on keyboards in the March issue of Computer is very interesting. I am sure we all agree that the present keyboard layout leaves much to be desired, but I cannot agree with his assumption that a completely nonmechanical keyboard would be faster and easier to use than one with tactile feedback.

The type of nonmechanical keyboards shown in Figures 3 and 5 of his article may offer some real advantages to the one-finger "hunt-and-peck" operator, but I cannot believe that a good touch-typing operator wouldn't be much faster with a good electromechanical keyboard.

With the keyboards described by the author, where does the operator rest his fingers? What happens if he leaves his fingers on the key positions? Without shaped finite keys, how does he know where his fingers are? Without the tactile feedback of a good electromechanical keyboard, how can he be sure that he has actually entered the character?

To operate Professor Montgomery's keyboard successfully, it seems to me, the operator would have to (a) look at the data to be entered, (b) look at the keyboard to locate the positions of the characters to be touched for entry, and (c) look at a display of some sort to make sure the character positions were properly touched and the desired characters entered. With a good electromechanical keyboard, the experienced operator can enter several lines of data without taking his eyes off the source document or looking at the keyboard, and he needs only glance occasionally at the display to check his entries.

Almost since keyboards were first used to enter data into computers (full-size, pocket, or personal), there have been two basic types of contact operation. One, which I call "click" or "toggle" action, requires a finite movement of a key to enter each character; i.e., to enter two successive 3's one must depress the "3" key until he feels it click on, let it up until he feels it click off, and depress it again. The other type uses a simple, direct, key-operated contact with a critical point at which the contact just closes, and the slightest tremor or vibration can produce multiple-contact closings or bounce. On some such keyboards a carelessly entered 3 may actually end up as a whole string of 3's! On the best of such units, careful electrical filtering may eliminate most of the bounce effect, but I have yet to see such a keyboard system that did not occasionally produce erroneous entries. And there is nothing in such action to positively indicate to the operator that he has depressed the key far enough to effect an entry.

For all these reasons, I would never willingly accept a keyboard without mechanical motion and the "click" type action.

Byron E. Phelps
Hendersonville, N.C.

Author's response

Editor:

The final writing of the article on keyboards omitted two important pieces of information. All working prototypes included a small speaker that provided a click that sounded very similar to the sound made by an old-fashioned mechanical typewriter. This immediate feedback is essential for a keyboard with no moving parts or one with membranous covers over a switch contact.

Knowing the location of the hands on the keyboard is also necessary. Several means for such orientation can be provided, even for a stylus-operated keyboard that is much smaller than a standard size.

I do not believe that the wipe-activated keyboard can be analyzed through modeling without measuring the interaction of various kinds of users with such a device. This can be done only through extended experience with a more perfected model than is now available.

Edward B. Montgomery
The University of Texas
Dallas, Tex.
Greatest-integer division

Editor:

This letter is prompted by Mr. Halasz' letter (Oct. 1981) titled "Usual Division Wrong?" Your readers might be interested to know that "greatest-integer" division was implemented in the GE/PAC 4040 (circa 1963) and subsequent 4000 series process computers produced by the General Electric Company until 1974. The rationale behind this selection was suggested by John Finnegan (last known to be teaching in Ireland). The writer had the fun task of selling it to GE engineering management.

The rationale had to do with performance. The 4040 had a serial 2's complement adder. Integer multiplication and division subroutines were slow. The greatest-integer algorithm required less fix-up—thus it was faster. Floating point was unthinkable. Certain applications demanded multiple precision. Greatest integer guarantees (as Mr. Halasz points out) that the second and subsequent words of a multiple-word quotient will be positive; they can be generated in order without affecting previous words of the quotient.

Greatest integer also turned out to be easier to implement in the GE/PAC 4050's discrete logic.

So much for rationale; the real purpose of this letter is to comment on user reaction to this unusual algorithm. Curiously enough, there was little feedback for several years. When asked, we pointed out that it was a "2's complement" divide consistent with the right shift. The term "greatest integer" was never used.

There were no serious objections until our users began using Fortran and found that certain programs behaved differently on GE/PAC computers. In due time, in order to conform to everyone else's Fortran, the compiler was modified to use a "divide-magnitude" subroutine; the next family member, the GE/PAC 4400, was enhanced with a new divide-magnitude instruction. Moreover, the original performance rationale was now moot—floating point was now commonplace.

It isn't always the case, but those who dare to be different can live to regret it.

Bob Erickson
Bethany, Conn.

June 1982

Developing procedures for code maintenance

Editor:

I would like to comment on the Ada letter by W. E. Drissel (January 1982 "Open Channel"). I have no experience or training in the use of Ada, so I cannot comment cogently on the truth or falsity of many of the assertions. I want to give a hearty "amen," however, to item 8, which suggests that most of the maintenance problems are in the heads of inadequate maintainers.

I have been maintaining many K-lines of ugly, unstructured (mostly Fortran) code for the last ten years. I have developed scientific procedures for the task, and they are effective enough that I have been asked to teach classes on the subject. I insist that maintenance costs are independent of the language used, once the programmers have thoroughly learned the language. Introducing a new language will only introduce a concomitant downtime while both designers and maintainers try to come up to speed on the new system.

G.R. Eugenia Schneider
Naval Weapons Center
China Lake, Calif.

On "Back into Reliable Software"

Editor:

Regarding Richard Ferrante's letter "Back into Reliable Software" in the September 1981 Computer ("Open Channel"), we have been developing ALFRED—Automatic Listing and Flowcharting Reorganized into an Elementary Dictionary. Using a Coral-66 program as input, ALFRED analyzes the Coral to form eight files of detail and a structured flowchart, which can be used to highlight changes between different issues of the program.

Further details are available upon request.

W. Honey
Plessey Marine
Wilkinthroop House
Templecombe
Somerset, England

Computer Architect

Stretch your resourcefulness in the new Bendix Advanced Technology Center in Columbia, Maryland... a very large-scale R&D operation that is charting the technological future of a more than $4-billion corporation.

Concentrating on basic and applied research and exploratory development, the Center can offer you one of the most formidable challenges any computer architect has ever faced.

We would like to hear from you if you have a Master's or PhD in Computer Engineering, Computer Science or Electrical Engineering and your related experience is particularly hardware-oriented—ideally including 2-5 years in the design of real-time control systems and advanced multiprocessor architecture. As our successful candidate for this most uncommon opportunity, you will be assigned increasing architectural hardware and software responsibilities. First and foremost, however, you will play a leading role in the design of a fault-tolerant multiprocessor for real-time control applications, helping to implement this architecture into VLSI.

To learn fully about the multiple advantages of joining our team of experts at our research facility located just 30 miles north of Washington, D.C., send your resume and salary history in complete confidence to: Mr. John C. Flato, The Bendix Corporation, Advanced Technology Center, 9140 Old Annapolis Road, Columbia, Maryland 21045.

Bendix Advanced Technology Center

An equal opportunity employer, m/f