Letters to the Editor

More on Jim Warren’s survey

Editor:
Jim Warren’s response (in the June issue) to my letter about the history of the Amateur Computer Society, and comments on my present position as editor of Creative Computing, “noted that the software published in Creative Computing has, to date, been limited to small programs written in Basic, sometimes reprinted from issues of People’s Computer Company.”

What Jim may not know is that only one program was ever reprinted from PCC, a music program in 8080 code, with permission and with attribution. Three other programs were submitted simultaneously by one author to both PCC and Creative. Since PCC has a slightly shorter lead time, the programs first appeared in print there, and later in Creative, so one might thus assume reprinting. But such was not the case. Sorry for the misunderstanding, Jim.

As for being limited to small programs, our Super Star Trek (May-June 1975) is 926 lines long, and Euchre (May-June 1977) runs to 561 lines. We try to print a diversity of programs—short ones so that readers with small systems can get them up and running quickly, as well as medium and large programs. A more fundamental issue is that we think that small Basic programs can be very valuable. For example, consider the large number of people who can learn about ELIZA (or DOCTOR) from our relatively short Basic version (Jul-Aug 1977) compared to the much smaller number who understood the original, lengthy Lisp version.

In other words, our objective is to expose the maximum number of people to new programming concepts in as clear and concise a way as possible.

Stephen B. Gray
Editor, Creative Computing

Concerning stack machines and the Burroughs B5000

Editor:
After reading David M. Bulman’s introduction to stack computers, which dwell heavily on the Burroughs B5000, I feel compelled to write to set the record straight concerning those who contributed to the design and architecture of that system.

From January 1959 to April 1961 I was the manager of the Burroughs Product Planning Group—consisting of approximately 20 professionals and located in Pasadena—that did the architecture of and specified the B200 and B5000 systems. Although the group was small in size, it had an extraordinary array of talent, including especially Paul King (as broadly talented a computer professional as there is in the business) and Jack Merner (a somewhat eccentric but exceptionally gifted programmer), both of whom were on my staff. In addition, we had Donald Knuth as a consultant when he elected to get his PhD from Cal Tech and Dave Dahm as a part-time employee or consultant. Bob Barton had already been working for Burroughs as the manager of a software activity. About the time I took over responsibility for the Product Planning Group, Bob converted from an employee to a consultant—a relationship which I believe is better suited to his temperament and working style—and became a consultant to our group.

Paul King was the manager of the B5000 project within Product Planning, and if the architecture of that system can be said to have evolved and matured in any single place, it would have to be Paul King’s blackboard. In addition to making major contributions to the system—some of which are delineated below—Paul also provided the necessary, but unpopular, filter function on Bob Barton’s ideas. Like many highly creative people, Bob has some very good ideas and some not very good ideas. The trick is to use the former and reject the latter. During the design of the B5000 some of Bob’s ideas made it across Paul’s blackboard and some did not.

Following are some of the major innovations (at least U.S. innovations) incorporated in the B5000, along with their source and the person or persons primarily responsible for them:

Virtual memory: In May 1960 UCLA conducted a two-week seminar entitled, “Using and Exploiting Giant Computers.” The program covered the IBM STRETCH computer, the Univac LARC, the Ferranti ATLAS 1 and Orison computers, the Bendix G-20, and a few other machines. The list of attendees shows that 14 people attended from IBM and seven from Univac. We sent Paul King and two design engineers from Burroughs. Paul and I have often mused that the 14 people from IBM were apparently so wrapped up in STRETCH that they failed to grasp the significance of what the late Stan Gill was saying about the virtual memory organizations of the ATLAS 1. Paul King did understand its significance and returned to Pasadena greatly excited about the concept. After a relatively brief period of review and discussion about how best to incorporate it, a segmented virtual memory was defined into the B5000 system (its project name in Product Planning at the time was the 4000 system). The credit for this first use of a virtual memory in a U.S. machine clearly lies with Paul King, not Bob Barton.

(It is worth noting at this point that the conceptual notion of a virtual machine had by this time already been a topic of much discussion around the Burroughs’ Pasadena facility. I believe the notion originated with Ted Glaser earlier in the 1950’s. Ted was in the Pasadena engineering group from 1956 to mid-1959.)

Several other B5000 design features can be traced to the May 1960 UCLA seminar. The idea of separate, modular input-output controllers can be traced to the LARC and the single number form can be traced to the G-20.

Polish notation: There is no question that the notion of producing a machine that directly operated on Polish strings and that used push down
operator/operand stacks was contributed by Bob Barton, as was the notion of contextual addressing.

**Algor based design:** The notion of designing the system to be efficient at handling a given language—Algor—was Bob Barton’s. Jack Merner, however, was our resident expert on the language and contributed most of the design ideas that made possible the execution of the concept. Donald Knuth, of course, consulted in this area.

**Procedure control:** Paul King came across a description of the PERM computer built by the University of Munich. It had a highly sophisticated method of subroutine control, including allowing them to be used recursively. Paul incorporated a more generalized version of the idea in the B5000, and it became the procedure control stack. Subsequently, Jack Merner suggested combining the two aforementioned stacks into a single stack concept, and this was done.

**Character manipulation:** Prior to the B5000 project (1959) there had been a 2111 computer design project in Pasadena. Under Ted Glaser’s architectural leadership the 2111 was probably the most sophisticated magnetic-strictive delay line machine ever contemplated. (The project was abandoned late in 1959, probably as a result of the 1401 announcement in October of that year.) Paul King had contributed the character string-manipulation capabilities to the 2111, and these were brought forward into the B5000. Several other B5000 innovations can be traced to the 2111. Among these are the notion of organizing the system around an exchange (we borrowed the idea from the telephone system), the notion of floating input-output channels, and the notion of I/O descriptors. (Contrary to David Bulman’s note, no mention of descriptors was made with the Atlas machine.) It is difficult to attribute some of these ideas to any particular person since they developed in a design discussion group that met weekly. The group included Ted Glaser, Paul King, Don Stevens, myself, and a number of others.

**Multiprocessor systems:** My own awareness of multiprocessor architecture came from reading an article on another German machine, the ER56. Whether this led to the B5000 being a multiprocessor system or whether the idea came from work done on military computers at Burroughs in Paoli is hard to say. Certainly the Paoli group contributed the notion of a conflict-resolving, switching-interlock system.

As can be seen from the above, the B5000 had a rich and varied ancestry. It certainly affirms the notion that good system architecture includes a lot of intelligent plagiarizing. The design of the B5000 system—like the design of any computer system—involves a number of major architectural contributions and hundreds of less major, but nonetheless significant, design contributions, sometimes involving the use of a single bit. Unfortunately, space does not permit mentioning all those who contributed or what they contributed. (I hereby ask some of my old friends to understand.)

The purpose of this letter is not to minimize the contribution of Bob Barton to the B5000 or to computer architecture in general. Rather, it is to place those contributions in perspective and to acknowledge that a number of other people—some not even mentioned in this letter—made significant contributions to the B5000. The total contribution of Paul King, in particular, was probably at least equal to that of Bob Barton.

Although the writer participated in many architectural and design sessions, he makes no direct claim to any of the major innovations in the B5000. I do, however, claim that I played the primary role in persuading the then pre-Ray MacDonald top management of Burroughs to proceed with and announce the system. Those were heady days in the computer field. It’s doubtful we will see their like again. I wonder if the top management of any computer company today, including Burroughs, could be persuaded to proceed with a system which included as many radical departures from current design philosophy. The reward for Burroughs’ gamble was a system, in the form of the B5000/5500/6700, which stayed in manufacturing for 10 years (probably the longest of any computer in the history of the field) and gave Burroughs a unique architectural-based position in the industry.

As a final point, it is worth noting that since my memory is no better than most, I refreshed it by rereading a fair array of detailed design notes and other material that exists from the period under discussion.

**Bulman replies**

Editor: I am grateful for the additional information about the B5000 project contributed by Mr. Lonergan. Before writing the survey, I talked with (among others) most of the people mentioned in his letter. When anyone was mentioned as having contributed significantly to the B5000 architecture, I attempted to locate and telephone them. Quite a chain was followed in this way. A significant fact which I should have thought to mention was that W. R. Lonergan was the individual who originally ‘sold’ the idea of building a stack machine to Burroughs corporate management.

As a result, there is little substantial disagreement between my outline of the project and Mr. Lonergan’s letter. Much of the apparent disagreement lies in the distinction which should be made between computer architecture and product definition. There is little doubt that many people contributed heavily to the definition of the B5000 as a product, including many in the Product Planning Group under Mr. Lonergan.

It is interesting to note that, as early as the summer of 1958, while working at Shell Research, Barton brought forward the idea that main storage should be allocated automatically by the hardware, rather than have the programmer concern himself with overlays from secondary memory. This certainly adds plausibility to the statements of essentially everyone else on the project that Barton was responsible for its virtual memory.

Another important idea of computer architecture is the use of the hardware stack for computational history (called procedure control in the letter). All the people from the project I talked with attribute most of this to Barton, with very significant contributions from Jack Merner. In addition to combining the two stacks, the much more important method of handling parameters called by name was invented by Merner.

I fervently agree with Mr. Lonergan that many other people contributed significantly to the success of the B5000 project. I only wish that I had been involved, so that I could share more deeply the amusement felt by those on the project, as they watch the features of their circa 1961 computer gradually being introduced by the major computer vendors of 1977.

W. R. Lonergan
Principal, Xerox

David M. Bulman
President, Pragmatics, Inc.