JUST SAY NO TO UNCLE SAM

MOST FEDERAL FUNDING for technology R&D is at best wasteful and at worst counterproductive, argues a noted technology consultant and innovator in this essay examining the government's proper role in research. There are areas where government largesse can be usefully applied, but these are limited.

BY *Gordon Bell*

The federal government has played a significant role in computer system development, including time-sharing, workstations, RISC architecture, computer networks and, most recently, parallel computers. Government agencies, especially ARPA (formerly DARPA), along with the Department of Energy, NASA, the National Science Foundation and others, have a remarkable record of nurturing the development of new computer structures. Artificial intelligence technologies (such as rule-based and expert systems programming and speech understanding, human interfaces and Unix as a standard) derived from ARPA-funded efforts are examples of such structures. Seymour Cray's invention of the supercomputer was made possible by the DOE's purchases and specifications.

But many of the government's funding efforts do little except perpetuate the marginal, social-welfare-like industrial research industry that exists today. A recent informal survey by New York–based Technologic Partners of recipients of federal dollars revealed them to be universally ungrateful, with little respect for their program monitors.

The Republicans now in control of Congress have talked about mini-

ILLUSTRATION BY PETE SPINO


ACME
SPINO

mizing research spending and giving it back to defense contractors, which support them. The Democrats want to have a bigger budget and give it to everybody, especially friends who support them. A third group, the bureaucrats and their grantees, are able to align with anything resembling political ideology (an oxymoron) to retain power and increase their budgets. Neither political party and no bureaucrats have an understanding of how to develop technology and turn it into an industry. They fail to see that risk-averse researchers are the last to understand the creation of companies, products or industries. Rather, these researchers are absorbed in writing proposals to maintain and expand a dependent research industry.

Most industrial-research or advanced-technology money goes down the drain. Take research funding for liquid-crystal display (LCD) technology. Companies that received this funding either lack the discipline to stop marginally useful programs or are in no position to commercialize technology or develop it into an industry. For instance, big bucks to Xerox Corp. for LCD panels follows the tradition of its recent computer research, with no channel for internal or external product commercialization. LCDs are likely to rival our last assault on massive parallelism. It is difficult to imagine Xerox or AT&T Corp., another funding recipient, ever becoming a supplier of LCD panels, whether they be high-volume, low-cost panels or specialized military products.

For the last decade, I've worked with and personally funded startup ventures that I think are competent and will make a difference. The assessment is really tricky, and I doubt if well-meaning bureaucrats or political appointees passing through Washington can get the hang of it easily. While the intentions of the Department of Commerce's Advanced Technology Program for Informational Technology (ATP) are laudable, the results are certain to be nil. The government could save the tens of millions of taxpayer dollars by eliminating this program. ATP's designation of its investment projects as "risky" might be better defined as, "The team doesn't know what the hell it's doing; to anyone competent, the projects are provably poor."

NINE POINTS OF LIGHT

Based on 30 years of observing government-funded technology research projects, here are my recommendations for creating a successful relationship between industry and government:

1 DEMAND-SIDE FUNDING that results in products that satisfy a customer and create a "buyer-seller" relationship is essential for every project. "Best effort" programs aimed at building knowledge or creating tools, products or processes are certain to be disastrous when it comes to computer systems R&D. Developing computer systems is unlike open-ended research and must be judged as development rather than research. All of us who have been involved in tools development understand this: Tools built in isolation by a "tools" group are rarely used or often break; the most useful tools are byproducts of an aggressive project that requires the tool!

2 DIRECT FUNDING of university or laboratory research for technology that is then carried over to start a company or become a product in an existing company is the most successful model for development and transfer of technology. Examples include Evans & Sutherland Computer Corp.'s graphics processors and Kuck Associates' parallelizing compilers. The many startups that came from Stanford University also show the efficacy of that form of transfer. Similarly, the University of California at Berkeley's NOW (Network of Workstations) project promises to be the right way to build massively parallel computers. It's unclear that many universities other than Stanford and Berkeley are qualified to develop computer systems.

The flow of technology from university research to companies is difficult and rare. The flow of California Institute of Technology's fine-grain multicomputers to Intel Corp. is a significant transfer to a company, even though many question the efficacy of the resulting product compared with Cal Tech's original. Carnegie-Mellon University's Warp failed to become successful at General Electric Corp., Honeywell Inc. or Intel, in part because those companies had no way to deliver and market the specialized products. The CMU Warp team and its successors, however, did become the founders of Fore Systems Inc. of Pittsburgh, the first startup to introduce an ATM switch.

3 COMPUTER SYSTEMS development has progressed rapidly using a demand-side approach—a tolerant customer (i.e., government) that buys high-tech products. Furthermore, the efforts of early users who codeveloped and evolved products have been critical for helping to create "industrial strength" products and companies.

ARPA-purchased facilities at universities are an example of effective use of this approach. Although time-sharing was invented at Massachusetts Institute of Technology, widespread use at universities was the key to its understanding and evolution. ARPA funded universities to purchase machines from Digital Equipment Corp., IBM Corp., Scientific Data Systems (a minicomputer company bought by Xerox) and Univac (now part of Unisys Corp.). Similarly, workstations from Symbolics Inc. and Sun Microsystems Inc. fostered artificial intelligence and distributed computing.

The approach that Sid Fernbach, former head of computation at Lawrence Livermore National Laboratory, used to help supercomputing come into existence was to specify needs, purchase equipment and be a knowledgeable, demanding and helpful customer. This was used successfully for supercomputers, high-speed networks, large file systems and large-scale, high-performance graphics terminals.

In contrast to Fernbach's methods, the Los Alamos National Laboratory's approach of funding its own engineers to develop special devices is unlikely to create anything except a home-grown product, and it is likely to be wasteful because technology transfer is so difficult. Products that a company acquires and builds based on a lab's technology and designs rarely succeed. Few, if any, startups or significant technology or products have come from Los Alamos.

Similarly, the supply-side or "build

it and they will come and program it" approach characterized by ARPA's Strategic Computing Initiative (SCI) or High-Performance Computer and Communications Initiative (HPCC) is much flawed and probably will fail to develop technology, products or lasting companies. Beginning with ARPA's funding of the Illiac IV, developed by the University of Illinois at Urbana, and continuing into the SCI in the early 1980s, state-sponsored computer efforts were decoupled from any real need.

For instance, the Illiac IV, eventually built by Burroughs, never worked well, nor did any of ARPA's user/researchers need it for their own work. The evolution of the CM1 from Thinking Machines Corp., also funded by ARPA, from a single-user computer without a floating point to finally having a 64-bit floating point illustrates the company's lack of focus and its groping for a use of its product. A traditional startup probably would have developed the proper features at the start and abandoned development efforts with little prospect of commercial application. Thinking Machines of Cambridge, Mass., filed for bankruptcy protection last year. Tera Computer Co. of Seattle, which is developing another ARPA-funded parallel machine, has failed to get any outside financing.

4 DIRECT FUNDING of large projects is risky regarding outcome, long-term training of personnel and establishing an industry. The early '70s example of ARPA's funding of Cambridge, Mass.-based Bolt Beranek and Newman Inc. (BBN) to develop packet switching for ARPAnet illustrates why direct product development and product purchase doesn't work very well. Perhaps the only way to develop ARPAnet was by funding its development by a single contractor, as ARPA did. I believe that an architecture and standard that allowed many suppliers to build equipment could have achieved even more impressive results. Ironically, this was exactly what happened when universities got involved in network research.

BBN became the sole source of packet switching for ARPA and the Department of Defense and eventually sold a few switches commercially. To increase its margins, BBN built high-priced, proprietary hardware that rapidly became obsolete. The market barely moved because BBN had a single customer—the military. Concurrently, DEC developed its own packet switching using ARPAnet ideas. With no ARPAnet implementers, the technology had to be redeveloped before products could be designed. In the 1990s computer networking is the hot area of the computing industry and the HPCC will further expand the market. Startups such as Cisco Systems Inc., 3Com Corp. and Bay Networks Inc. have grown to have revenues exceeding $1 billion. Many communications companies are staffed with BBN and DEC alumni. BBN is a minor supplier.

> STATE-FUNDED COMPANIES CAN USE PROFITABLE GOVERNMENT CONTRACTS TO SUBSIDIZE THE COMMERICAL PART OF THEIR BUSINESS, THEREBY CREATING AN UNFAIR AND ARTIFICIAL MARKET.

5 DIRECT FUNDING of product development within companies, as practiced in ARPA's SCI and HPCC, is wasteful and likely to impede the development of technology by having to feed an overpopulated industry. In the early '80s, ARPA funded roughly 20 parallel computer efforts that initiated the SCI. None were successful. State funding doesn't build a strong company because it provides a safety net that is not enduring. Furthermore, state-funded companies can use profitable government contracts to subsidize the commercial part of their business, thereby creating an unfair and artificial market. Since such companies have a monopoly within the leading-edge, early adopter university and laboratory market, privately funded companies are denied the key to market entry.

In the case of the HPCC, the traditional, fragile supercomputer market is disturbed and may be destroyed by the hype about massive parallelism backed by government-supported computer purchases. There is little evidence to support massive parallelism as a superior alternative to traditional supercomputers or clusters of workstations, which would have occurred in an evolutionary fashion. In fact, the more we learn about the actual performance of state-sponsored computers, the poorer and more highly specialized they look. Their principal value is in training students. Another benefit is software that can be used on the workstation clusters that should have been supported.

The technical computing market has always attracted entrepreneurial engineers. In the '90s the likelihood of starting a successful parallel computing company is small because of an overcrowded market. Competent, established companies such as Convex Computer Corp. of Richardson, Tex., and IBM are working on massive parallelism using workstations to further reduce prices to the commodity level. Also, many computer companies are using ATM, which will permit distributed, coarse-grain computer clusters.

6 UNIVERSITY-COMPANY collaboration is a new and worthwhile area of government R&D. Any company should be free to work with a government-funded university project or laboratory to produce technology or product prototypes. In fact, companies should be especially encouraged to collaborate through tax incentives. Researchers should consider industrial partners based on their ability to market the product and avoid funding

those that don't have the requisite market presence. For example, Burroughs (now part of Unisys), GE, Honeywell and Motorola Inc. have been especially inept partners for computing because they lack adequate market presence in the customer and application sectors, where new computer structures have been most relevant.

A company should not be government-funded as part of a university-company collaborative product development. If a company spends its own funds, it is more likely to take the technology and product seriously as part of its product portfolio, and not as a way to recover its research costs or pay for marginal people or work.

7 CRADAS (cooperative research and development agreements) are closely allied to direct product development. CRADAs have the advantage of being driven by customers or laboratories that can either supply technology or buy a product. In many cases, a CRADA is a thinly disguised purchase order in which an agency gives money to a laboratory to buy a computer and fund laboratory and company product development. The company provides window-dressing funds that would have previously been considered as a discount on the resulting product when purchased by the government.

8 FUNDING OF applications development is the most significant way to fund and subsidize a given computer system development because equipment purchase and use implies an endorsement. Industrial and other partners are implicitly committed to purchase the same or compatible system. The only acceptable way to fund applications development is through the porting and benchmarking of a set of computers. This facilitates the training, understanding and use that would otherwise be missing.

9 THE GOVERNMENT should discourage designs from those who have not used or built a successful computer. The world is drowning in massively parallel computers that absorb programmer time chasing peak announced performance. Furthermore, the number of computers (and options) will increase over the next decade as Convex, IBM, Japanese manufacturers and others enter the market. Universities capable of system and applications development are essential to the software effort implicit in massive parallelism.

WHAT WORKS, WHAT DOESN'T

To summarize, only two methods of government support have been effective: funding university research that is transferred via a startup company and encouraging knowledgeable, early-

> IF A COMPANY SPENDS ITS OWN FUNDS, IT IS MORE LIKELY TO TAKE THE TECHNOLOGY AND PRODUCT SERIOUSLY.

adopter grantees to purchase advanced systems to validate them and assist in their development and evolution.

In many of the successful cases of technology transfer, technology in the form of a product prototype was developed in a university. The technologist left the laboratory, got financing, and went on to develop a product. The university research, government laboratory and leading-edge industrial communities acting as early adopters bought, applied and helped evolve the product. DEC was formed in this model by building on technology from MIT's Lincoln Laboratory. Similarly, government-supported Engineering Research Associates begot Control Data Corp., which begot Cray Research and Cray Computer. For a score of years beginning in the early 1960s, DOE's Livermore and Los Alamos Laboratories specified, bought and applied Crays, creating the supercomputing industry.

Sun Microsystems came from a prototype and team from Stanford and Berkeley universities. Cisco, Mips Computer Systems Inc. and Silicon Graphics Inc. came from Stanford. With the availability of capital for competent teams, and the drive and skill for success, entrepreneurial startups will continue to be the main way to form lasting, successful companies. However, immortality is not guaranteed (witness Control Data, Cray Computer and DEC).

Direct government funding of a company to develop a product, followed by funding product purchases, doesn't appear to produce healthy companies. France's Advanced Computer Research Institute went into liquidation in April. In the United States, such state-ordained computer companies (i.e., Cray Research, Tera Computer and Thinking Machines) are likely to impede the natural evolution of technology and product development.

ARPAnet is probably the most successfully funded product development, but it can be argued that funding BBN to develop ARPAnet impeded the wide-scale adoption of packet switching because of BBN's choice of proprietary hardware and the absence of pressure to reduce costs. Furthermore, state-sponsored products create an artificial market and nonlevel playing field that deny privately funded companies the early adopter market characterized by universities and government laboratories. ■

---

*Gordon Bell, a Los Altos, Calif.-based technology consultant, spent 23 years at Digital Equipment Corp. as vice president of R&D. He established the computing directorate at the National Science Foundation and coauthored the High Performance Computing and Communications Initiative. In June he won the MCI-*ComputerWorld-*Smithsonian award for innovation.*

For Bell's analysis of 32 government-funded technology projects, click on the UPSIDE icon in eWorld's Computer Center, or see UPSIDE's Internet home page at http:/www.upside.com/upside/upside.html