

# The Innovation Engine: Government-Funded Academic Research

*David A. Patterson*

Electrical Engineering and Computer Sciences  
University of California, Berkeley

Technical Report No. UCB/EECS-2025-140

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2025/EECS-2025-140.html>

July 4, 2025



Copyright © 2025, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

### Acknowledgement

A paper like this makes one reflect on their academic careers. As proud as I am of our research accomplishments, it's the personal relationships that one treasures. Thanks go to the hundreds of students and postdocs who worked hard to realize the visions of these labs, many of whom have made large marks of their own. The labs also relied on dozens of long serving staff. And special thanks to the dozens of faculty colleagues whom I was honored to work with over four decades.

# The Innovation Engine: Government-Funded Academic Research

David Patterson, Pardee Professor Emeritus, UC Berkeley

July 4, 2025 (to appear in *Communications of the ACM*)

*Abstract: Five government-funded academic research projects over one career at one university delivered economic benefits across nearly 90% of U.S. states with new product sales about 10,000 times the US government funding—returning taxes back over 1000 times its investment—and trained generations of innovators.*

## Introduction: Five home run projects as case studies of GoFAR

Government-funded academic research (GoFAR), lately the subject of across-the-board cuts in the US, is one of the engines that truly makes America great. When I started as a new assistant professor in 1976, I was advised to aim my research objectives for “home runs”. In an environment in which ambitious projects with high potential weren’t penalized if they fell short, my colleagues and I strove to make a high impact. The thinking was that you’re more likely to hit home runs by swinging for the fences than by bunting for singles. And so we swung. NSF and DARPA used grants, contracts, and fellowships to sponsor our research, alongside some smaller donations from industry.

Here is a quick summary of five highest impact projects—which Table 1 summarizes and the second half of the paper details—of the 11 that cover my academic career:

- **Reduced Instruction Set Computer (RISC)** [24]. Simplified instruction sets enabled faster microprocessors. Today, 99% of all computers follow RISC principles; the R in the ubiquitous ARM architecture—with nearly 300B chips built—stands for RISC.
- **Redundant Array of Inexpensive Disks (RAID)** [29]. Strength in numbers via redundant arrays of small disks provide better cost, performance, and reliability than large disks, leading to >\$100B in sales.
- **Network of Workstations (NOW)** [2]. Networked clusters of off-the-shelf workstations laid the foundation for Internet service infrastructure, whose descendants power hyperscalers worth trillions today.
- **Reliable Adaptive Distributed Systems Lab (RAD Lab)**. Machine learning combined with systems expertise led to the Spark analytics engine [39]—that 40,000 companies use—and the startup Databricks.
- **Parallel Computing Lab (ParLab)** developed RISC-V [37], an open instruction set architecture that any company can use for free. In 2024, 2B RISC-V chips shipped, projected to grow to 20B by 2031.

Lab	RISC	RAID	NOW	RAD Lab	ParLab
When	1980-1984	1988-1992	1993-1998	2005-2011	2007-2013
Key result/artifact	RISC-I and RISC-II microprocessors	RAID-I and RAID-II storage servers, Postgres DBMS, Log structured file systems	NOW-I and NOW-II clusters, Inktomi search engine	Mesos cluster manager, Spark data analytics system	RISC-V open architecture, Roofline performance model, Selective embedded just in time specialization
Startups	MIPS, Pyramid Technology, Ridge Computers	Array Technologies Corporation, NetApp	Inktomi, Scale 8	<a href="#">Comfy</a> , Databricks, Mesosphere	AheadComputing, Akeana, Bina Technologies, Codaip, Condor Computing, Cortus, FuriosaAI, Rivos, SiFive, Tenstorrent, Ventana
Companies influenced	ARM (Advanced RISC Machines), Sun Microsystems, HP, & many others	DEC, EMC, Hitachi, IBM, NCR, StorageTek, Sun Microsystems, & many others	Google, HP, Sun Microsystems, & many others	Amazon, Cloudera, Google, Microsoft + 40,000 companies that use Spark	Andes Technology, MIPS, NVIDIA, Qualcomm, Samsung, Synopsys, & many others
Open Source/Open Standard	<a href="#">Magic</a> ECAD tool	<a href="#">Postgres</a> DBMS	xFS cloud file system	<a href="#">Apache Spark</a> , <a href="#">Apache Mesos</a>	<a href="#">RISC-V specification</a> , <a href="#">Rocket chip Open HW</a> [5], GPUSVM
Awards	ACM dissertation award; ACM Turing award; ACM/IEEE Eckert-Mauchly award; IEEE von Neumann medal; NAE Draper prize	2 ACM dissertation awards; IEEE Johnson storage award; 4 Test of time / Hall of fame awards (1 SIGMOD, 2 from SIGOPS, 1 IFIP 10.4)	1st cluster in Top 500 supercomputer list; 3 Graysort records; 1 Test of time award (HPDC); 2 Best paper awards (SOSP, Hot Interconnects)	ACM dissertation award; ACM Weiser award; Frontiers of science award; 5 Test of time awards (ICDE, ICML, NSDI, SIGCOMM, SOCC)	ACM Athena award; ACM Ken Kennedy award; IEEE Charles Babbage award; 2 Best paper awards (IPDPS, SPAA), 1 Test of time award (DAC)
Highest Cited Paper, # Citations	[24], 748	[29], 4742	[2], 1426	[3], 15034	[4], 3172

**Table 1. Impact of five “home run” research projects over my academic career (1976–2016). Companies in this table have (or had) offices and employees in 44 of the 50 U.S. states, where 98% of Americans live.**

The companies in Table 1—with offices and employees across 44 of the US states<sup>1</sup> that hold 98% of the total US population—sold hundreds of billions of dollars of products based on these breakthrough technologies from GoFAR, which also found homes in thousands of military systems that enhanced national security.

Eight principles guided these home-run projects, which were structured as “labs” reflecting the collaborative team of students and faculty in a shared space (see [26]):

- **Multidisciplinary** teams of three to five faculty experts in different fields, as there are more chances for impact across fields than within individual fields.
- **Demonstrative usable artifacts** require genuine cross-discipline collaboration that leads to breakthroughs. Rather than toy demos whose goal is to produce papers, they are realistic enough to win over skeptics to help technology adoption.
- **Seven to ten year impact horizons** instead of distant futures. In our incredibly fast-moving field, no one can see accurately 15 to 20 years ahead.
- **Five-year sunset clauses create urgency** and allow for new opportunities. Many five-year labs over a career give you more chances for home runs than a few 10- or 20-year projects. Deadlines are rare in academia, so the timeline also gives a real target by which to demonstrate lab goals. It is easier as well to get commitments from experts in several fields to collaborate for five years than for longer.
- **Sunset clauses lower opportunity costs.** It takes a decade to determine the level of a lab’s success. Most are not a home run; six of my other projects did not make Table 1. But any good entrepreneur knows that you don’t get breakthroughs without risk and the possibility of failure. Without a time limit, projects can linger until everyone loses interest, a potentially enormous opportunity cost. Sunset clauses help researchers and funders move on to the next promising project.
- **Biannual three-day offsite retreats offer regular honest feedback**, provide deadlines, and build team spirit. The most important feature is the praise and constructive criticism in the last session from external practitioners and researchers, which are deeper, more thorough, more thoughtful, and more frequent than most paper reviews.
- **Physical proximity** of collaborators, ideally in one large physical space. Multi-university projects are less successful while multi-disciplinary projects at a single university excel [15].
- **Leadership focused on team success** rather than individual recognition. I led about half of the 11 labs over my career, and was happy to have colleagues lead others. The leader builds team spirit, focuses work towards lab goals, and allows delegation of lab adminstrivia to a benevolent decision maker.

## Why should the government partner with academia?

After World War II Vannevar Bush argued that investing in scientific research at universities would have a tremendous return to the economy, to healthcare, and to national defense [9]. The goal is for government, academia, and industry to be synergistic partners, all playing to their strengths. Eighty years later, here are my top ten reasons why this GoFAR partnership has been so effective:

1. **Universities publish their research results**, whether it succeeds or fails, so that everyone can learn from the effort. For example, RISC papers inspired ARM and RAID papers encouraged EMC to produce successful products without any direct contact with the original inventors (see below). Most companies have no such tradition.
2. **Successful university research projects can lead to new companies.** Over my career there have been numerous examples of university research projects that led to startups that grew to be major corporations. It can be challenging to form a successful startup spun off from a large company when it owns the intellectual property.
3. **Multidisciplinary research is an inherent strength of top universities.** Where else can one gather experts from all areas of science, engineering, arts, and so on in one location and have them talk to each other? Research at the intersection of computer science and other disciplines has led to advances that would have been difficult to achieve in industry, where groups are often more narrowly focused. For example, top experts in computer science and the top experts in neuroscience collaborated to read minds from MRI data [23].
4. **Academic freedom enables exploration of unconventional ideas.** Academic researchers have the intellectual freedom to pursue high-risk, high-reward ideas that may not have immediate commercial applications. This exploratory nature can lead to unexpected paradigm-shifting breakthroughs. For example, RAID (described below) was a byproduct of curiosity-driven research. Industry, in contrast, is typically more risk-averse and goal-oriented, understandably more focused on deliverables with short- to medium-term payoff.
5. **It can be awkward for companies to develop and adopt new technologies that disrupt current product lines.** An academic’s sole concern is advancing the state of the art. For many of the home run labs in Table 1, the leading companies in the area were the last to embrace the innovation, as they were highly profitable in the current marketplace and had little desire for change. Christensen refers to this as *The Innovator’s Dilemma* [12].

---

<sup>1</sup> Companies in Table 1 have or had employees and offices in Alabama, Arizona, Arkansas, California, Colorado, Connecticut, Florida, Georgia, Hawaii, Idaho, Indiana, Iowa, Illinois, Kansas, Kentucky, Louisiana, Maine, Maryland, Massachusetts, Michigan, Minnesota, Missouri, Nebraska, Nevada, New Hampshire, New Jersey, New Mexico, New York, North Carolina, Ohio, Oklahoma, Oregon, Pennsylvania, Rhode Island, South Carolina, Tennessee, Texas, Utah, Vermont, Virginia, Washington, West Virginia, Wisconsin, and Wyoming.

6. **Industry has reduced the amount of internal pure research it funds**, especially for high risk research that is the *raison d'être* of the high-impact labs. Computer science research labs in industry that played vital roles in the 20th century are shadows of their former selves. Almost all company's R&D in this century is advanced product development. Our society relies much more on GoFAR for foundational research today than when I started my career.

7. **Top universities attract top people from around the world.** Academia attracts very bright people worldwide who want to earn advanced degrees. Some who receive advanced degrees become leaders of existing corporations (e.g., AMD, Google, and Microsoft) or found new ones (e.g., Hewlett Packard, Intel, NVIDIA, Netflix). Ensuring that US universities can draw from the brightest of the global population of 8 billion—25 times larger than the US domestic population—has been and is vital to US success in science and engineering

8. **Even projects that are not home runs train students.** After graduation they can become innovators and make their own contributions. While universities produce novel ideas and transfer technology to existing companies and to startups, their most important product is people [25]. The silver lining of research projects in industry that do not pan out is unclear.

9. **Computer science is a young person's field.** Our technology changes so rapidly that the state of the art 20 or even 10 years ago can be nearly irrelevant today. A student may have better knowledge of the most critical material than someone with decades of experience. Having a research project staffed by brilliant, hard-working, up-to-date, young people with less experience is not necessarily a huge disadvantage in computer science.

10. **Funds go much further at universities than in industry.** Faculty salaries are lower than industry, plus research funds pay only a small part of faculty salary—primarily for summer support—and students receive a much smaller salary compared to industry employees. Industry overhead, commonly greater than 100%, is roughly double the university overhead on the GoFAR grants in part due to more layers of management to pay for and in part since universities partially subsidize it. This lower cost means GoFAR can explore more topics for the same investment.

## GoFAR funds primarily support students

Most of the budget of a GoFAR grant is for the students doing the work, not for equipment, staff, or faculty. Such labs lead to a major positive impact on technology and the industry, and tremendous training grounds given their team orientation and collaborative multidisciplinary goals. Renowned computer architect [Burton Smith](#) called the Par Lab team in the 2010s “the best group of Ph.D. students that I have ever seen”, which echoed computer visionary [Mark Weiser](#)'s comment almost word for word in the late 1980s about the fourth RISC project. These labs also uncover treasure troves of Ph.D. topics, as the ACM dissertation awards in Table 1 attest.

Beyond technical innovations, a byproduct of these labs is future leaders of our field. Lab alumni have gone on to found billion dollar startups, become technical leads of large corporations, and successful researchers and leaders at top universities.

## Exploring five GoFAR home runs

Next are more detailed histories of my five home run labs and their impact, starting with RISC [27].

### #1: Reduced Instruction Set Computer (RISC)

When Stanford's John Hennessy and I were assistant professors in 1980, conventional wisdom held that computer instruction sets—the vocabulary that software uses to talk to hardware—were too low-level, burdening programmers and causing software failures. The trend was toward complex instruction sets to bridge the gap between people and machines.

In the 1970s, microprocessors were only found in home appliances. We believed microprocessors would become computing's foundation, following Moore's Law of doubling transistor counts every year or two. The question was: what instruction set would best serve these rapidly improving microprocessors?

The success of the operating system UNIX, written in a high-level language, changed perspectives. The issue wasn't programmers writing machine code anymore even for operating systems, but whether compilers could produce efficient programs for an instruction set.

Together with my colleague Carlo Séquin, we argued for a *Reduced Instruction Set Computer (RISC)*, keeping instructions simple rather than complex [28]. We termed the conventional approach *Complex Instruction Set Computers (CISC)*. We believed RISC would be easier to build and easier for compilers to utilize.

The debate centered on performance: while CISC might require fewer instructions due to their sophistication, each instruction might take longer to execute than a RISC instruction—like a page of polysyllabic words potentially taking longer to read than simpler words.

I sent a draft of the case for RISC paper to friends in industry building CISC minicomputers. Instead of sending me comments, they wrote a rebuttal to appear next to our paper [13].

This scientific question became emotionally charged in the computer design community. CISC advocates believed RISC would complicate software; RISC advocates argued compilers could hide these details from programmers. Despite our universities' on-the-field athletics rivalry, John Hennessy and I joined forces to advocate for RISC.

Industry debates grew heated at conferences beginning in 1982. Similar discussions had occurred earlier at IBM around the 801 project led by John Cocke concerning minicomputers, though IBM management delayed publicly sharing their views until later.

Research ultimately showed that while RISC needed about 30% more instructions, it processed them approximately 5 times more quickly, making RISC 3-4 times faster overall. Additionally, RISC microprocessors required less hardware and power—a crucial advantage as computing became mobile and battery-powered.

In 1983, Hennessy's Ph.D. students including Chris Rowen and Norm Jouppi and our Ph.D. students Robert Sherburne and Manolis Katevenis presented their RISC microprocessors at the major microchip conference, stunning the audience by creating designs arguably superior to industry state-of-the-art [31][32].

Max Planck said that scientific truth does not triumph by convincing opponents and making them see the light, but that science advances one funeral at a time. Computer architecture benefits from the commercial market that tests new ideas, so we don't have to wait for funerals to change the field.

For example, in 1983, Cambridge-based Steve Furber and Sophie Wilson created a new microprocessor for the Acorn personal computer. Inspired by our RISC papers, they developed the Acorn RISC Machine (ARM) with two "advantages": no money and no engineers. These constraints prioritized simplicity, aligning perfectly with RISC philosophy. The ARM1 debuted in 1985 as the first commercial RISC processor, outperforming all microprocessors in the market.

Apple approached Acorn in 1990, interested in ARM for its new Newton handheld device. Only RISC could meet Newton's performance, power, and cost requirements. Acorn agreed to Apple's request to spin off ARM as a joint venture, rebranding it as Advanced RISC Machine. While the Newton failed commercially, ARM's efficiency made it ideal for cell phones. At that time, Nokia was the leading supplier of cell phones, so the selection of ARM for the Nokia GSM phone (global system for mobile communications) in 1998 was a major boost. The Nokia experience helped ARM understand system-on-chip requirements, positioning it to dominate the smartphone and embedded computing revolution for the following decades.

With almost 300 billion ARM chips shipped—nearly 40 per person globally—99% of processors today are RISC-based, which traces its roots back to GoFAR. RISC's simplicity was more efficient in silicon use and power consumption, driving its success. Beyond providing faster and more economical computing for the world, RISC generated substantial economic benefits through job creation and tax revenue.<sup>2</sup>

## #2: Redundant Array of Inexpensive Disks (RAID)

This project started with a question. My colleague Randy Katz was an early Macintosh user, which when announced in 1984 relied on floppy disks for storage. A few years later the first small hard disk drive was developed for Mac and the IBM PC, which was a godsend for personal computer users. Randy's question was "I wonder what else we could do with these small disks?" His curiosity sparked a revolution in storage.

At the time, hard disk drives for mainframes were the size of dishwashers and the size of microwave ovens for minicomputers. Katz, our Ph.D. student Garth Gibson, and I speculated that we could replace one mainframe hard drive with 100 small PC drives (and one minicomputer drive with 10). We wrote a draft showing that it would be cheaper for the same capacity but have much greater performance given we had 100 drives accessing data versus one large drive, even if one large drive was much faster than one small PC drive. We sent it to a friend at IBM. His feedback highlighted our critical oversight: reliability. With 100 drives, failure rates multiplied dramatically, as it was at least 100 times more likely that one drive would fail and we would lose data. This insight inspired us to add redundancy—and to name our project RAID for *Redundant* Array of Inexpensive Disks.

We found some related projects offering reliable storage via redundancy at other companies. To clarify our work, the paper that debuted RAID identified five levels of sophistication of redundancy that had increasing benefits in cost-performance. Digital Equipment Corporation (DEC) and Tandem Computers offered RAID level 1, Thinking Machines sold RAID level 2, IBM filed a patent on RAID level 4, and we built a prototype of RAID level 5 we dubbed RAID-I. RAID-II was next, which had 144 disk drives under a single storage controller attached to a high speed network. RAID-II is now housed at the Computer History Museum.

Our bottom line demonstrated strength in numbers; a RAID system was about 10 times better in cost/performance/reliability than a mainframe drive and about 5 times better than a minicomputer drive. When the paper was published [29], it received immediate attention. It even led to a tutorial in a magazine for personal computers [1], which was a much different market than we expected to show interest.

The paper also landed on the desks of executives at EMC Corporation, who were facing a crisis. Their main product was cheaper memory modules for IBM mainframe computers. They had recently been squeezed out of that market due to changes at IBM, and needed a new product. EMC decided to embrace the RAID ideas to offer reliable storage for IBM mainframe computers using arrays of more cost effective smaller drives, saving the company.

After discussing with Katz, marketers later changed "Inexpensive" to "Independent" in the RAID name for pricing

---

<sup>2</sup> When I asked ChatGPT to quantify economic impact using this paragraph as a prompt, it replied "RISC has enabled millions of jobs globally and contributes upwards of \$100B annually in economic and tax value, directly and indirectly. It's one of the most economically impactful architectural innovations in tech history." Claude and Gemini gave even rosier forecasts.



flexibility. The numbered RAID levels had an unintended consequence: companies invented higher levels (beyond 5) to suggest superiority, sometimes with technical merit but often as marketing plays. These events helped inspire companies to form the RAID Advisory Board to advocate and evolve RAID technology [21].

GoFAR-based RAID became tremendously successful, with EMC alone generating \$25B in seven years, suggesting industry-wide revenues exceeding \$100B. Two associated projects also achieved significant impact: RAID's limited write performance inspired the development of *log-structured file systems* [30], now foundational in many products; and the open source POSTGRES object-relational database [34], which evolved into PostgreSQL, the open source database used by thousands of companies worldwide.

### #3: Network of Workstations (NOW)

Some supercomputers were built using many processors going back at least to the 1980s [16]. In 1995 we proposed building more cost-effective supercomputers using off-the-shelf workstations connected via emerging switch-based networks like Myrinet [7]. Our Network of Workstations (NOW) project [2] competed philosophically with Stanford's DASH [20], which bet on cache-coherent memory in large scale multiprocessors to simplify parallel programming. This rivalry between what became known as clusters versus large scale shared memory processors was a popular discussion topic in computer architecture circles in the 1990s.

In April 1997 NOW set two sorting records. That same month NOW demonstrated versatility by becoming the first cluster ranked in the [Top500 supercomputers list](#). It is rare for the same hardware to be great at both data processing and number crunching. Despite some supercomputers lasting six years in the Top500, within five years 20% of the world's fastest computers were clusters and in ten years 90% of the new entries and 80% overall were clusters (Figure 1).

We also developed xFS, a precursor of cloud file systems [36].

However, the most influential application proved to be Internet services. While AltaVista—the leading search engine in the late 1990s—ran on large scale shared memory multiprocessors, new assistant professor Eric Brewer and his Ph.D. student Paul Gauthier recognized clusters offered better cost/performance, scalability, and fault isolation.

Their Inktomi search engine, built on NOW principles, became more popular than AltaVista despite running on a university campus. They then started an eponymous company that dominated search until Google's debut. Google and many other companies followed Inktomi's lead of delivering Internet services on such clusters of many inexpensive computers. Today, descendants of these GoFAR clusters in hyperscaler data centers are the computing foundation for companies worth trillions of dollars employing hundreds of thousands.

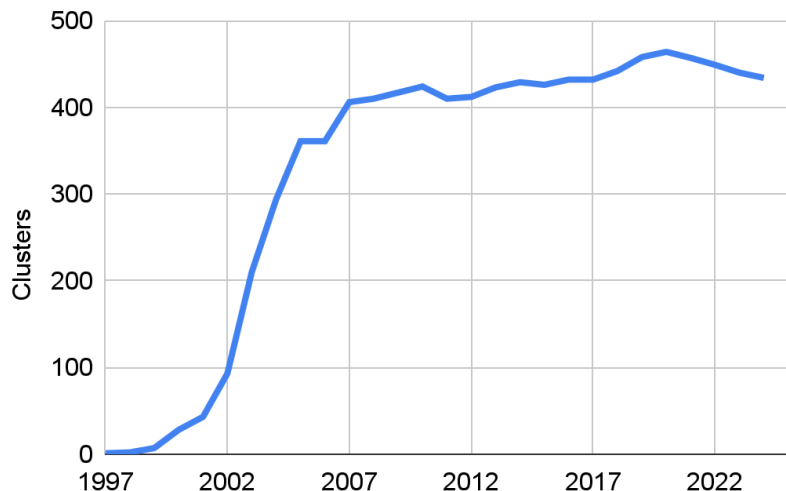


Fig. 1. Number of clusters in the Top500 supercomputer list.

### #4: Reliable Adaptive Distributed Systems Lab (RAD Lab)

The Reliable Adaptive Distributed Systems (RAD) Lab combined machine learning with systems expertise to enable rapid development of revolutionary Internet services and for data centers to become self-healing and self-managing. Pursuing this agenda led us to become early Amazon Web Services customers. Our workloads tested early cloud scalability limits, positioning us to author a definitive vision paper explaining cloud computing's importance and research directions to improve it [3].

It was also the first lab to face the problem that computers and the Internet were just as fast at home as they were on campus. Students and faculty started working more from home to avoid interruptions. Our solution to combat isolation was remodeling workspace to encourage on-site collaboration with free drinks and attractive meeting rooms [26]. The surge in productivity from spontaneous interactions outweighed occasional interruptions.

In this collaborative open environment, systems Ph.D. students sat near machine learning Ph.D. students struggling to scale their algorithms using MapReduce. This cross-disciplinary interaction inspired Matei Zaharia to create Spark [39], an efficient programming system for data center-scale machine learning and other algorithms. Spark is now used by over 40,000 organizations worldwide. A few years later some graduate students and faculty from the RAD Lab founded Databricks, starting from Spark, which now employs over 7000 people.

### #5: Parallel Computing Lab (Par Lab)

The Parallel Computing (Par) Lab was a multidisciplinary research project exploring the future of parallel processing inside a microprocessor. We projected a future with thousands of cores per chip [4]. Today server microprocessors have hundreds of cores and GPUs support thousands of hardware threads. We developed the roofline performance model [38]—widely used to improve designs of both parallel hardware and parallel software—

and one of the earliest demonstrations that GPUs were also a good match to machine learning. GPUSVM is an open source program that novelly used GPUs to deliver ~100x improvement over conventional machine learning libraries running on CPUs [10]. All popular programming frameworks for AI—Tensorflow, JAX, Pytorch—are built on Python using *Selective Embedded Just In Time Specialization (SEJITS)* to create kernels for parallel computers, which the ParLab also pioneered [11].

For our hardware research, Krste Asanović proposed that we create a new, clean RISC instruction set architecture leveraging what was learned since the 1980s. The goal was an instruction set architecture that we could use both in our classes and in our research. Following departmental traditions, we made it an open standard in the hopes that other academics would use it in their research and thereby form a community to share software development and other tools. Ph.D. students Andrew Waterman and Yunsup Lee led the design and built RISC chips. To honor the four generations of our RISC chips in the 1980s, they named it RISC-V (“RISC five”) [37].

A few years later we discovered a desire for a free open instruction set standard that any company could use. Surprisingly, industry—not academia—embraced RISC-V. We then did whatever we could to make RISC-V succeed in that role. Following the 1980 RISC playbook, we wrote a position paper for open ISAs and invited authors at ARM to write a rebuttal. For many, their first introduction to RISC-V was this debate inspired by “The Case For Open Instruction Sets” [6] versus “The Case For Licensed Instruction Sets” [33]. A few months later, we held the inaugural RISC-V workshop and started a non-profit foundation to popularize and evolve RISC-V.

The vision of a universal computing instruction set standard available to everyone without charge inspired an almost religious fervor for RISC-V [18], similar to that for the open-source software in the 1990s. Today, RISC-V International includes over 400 organizations with annual summits across three continents. Two billion RISC-V chips shipped in 2024—projected to reach 20 billion by 2031—and it spawned a dozen startups.

Once again, high impact GoFAR created billions in economic value and numerous jobs.

## If GoFAR is not broken, why change it?

History shows GoFAR works. The home-run labs in Table 1 are just those associated with a portion of the faculty in one department at one university. A 25-year series of reports from the National Academies documents the proud track record of the symbiotic relationship of computer science researchers, government, and the computer industry [8] [17] [19] [22]. They created a “virtuous cycle” where new technologies inspire new research, which creates new ideas and new technologies, over and over. Examples of these computer science breakthroughs from this virtuous cycle that benefited the US economy and defense of the US include:

- Broadband and mobile in digital communications,
- Microprocessors,
- Personal computing,
- The Internet and the World Wide Web,
- Cloud computing,
- Database management systems,
- Computer graphics for design and entertainment,
- AI and robotics.

Of course, GoFAR discoveries are not limited to computer science: CRISPR gene editing, the Human Genome Project, lasers, mRNA vaccine technology, MRI (magnetic resonance imaging), PET (positron emission tomography), and solar panels are just a few extrinsic examples of many. Imagine what the world would be like without such innovations that were driven in part by GoFAR at dozens of universities.

## Conclusion: A thousandfold return on GoFAR investments

My best estimate of the total funding from the US government for all 11 projects I was involved in over my 40 year career—that supported hundreds of students and dozens of faculty—is less than \$100M, even accounting for inflation. Much more than \$100B of products were shipped based substantially on technology developed in these GoFAR labs, and likely over \$1000B in today’s dollars. Note that this metric doesn’t account for the economic benefits of contributions from lab alumni over their careers.

The ratio of product sales to government research investment would then be roughly 10,000:1. Beyond whatever financial benefits innovative products provide to society, an organizational self-interest question is how much is directly returned to the government in terms of taxes on these products. There is no single definitive number, but a conservative estimate is that >10% of sales go to the federal government in taxes.<sup>3</sup> If so, the direct tax return to the federal government would be at least 1000 times its research funds that supported our 11 five-year labs.

---

<sup>3</sup> Tax overhead varies because corporate income taxes are based solely on profits; income taxes paid by employees and employer payroll taxes are a function of labor intensity of the product and the level of employee wages; taxes paid by suppliers and import duties are a function of the supply chain; and sales and property taxes depend on where the products are purchased. One way to estimate is that total federal revenue is typically 17–18% of the US gross domestic product [14]. The average lower bound estimate from ChapGPT, Claude, and Gemini was 13% of product sales, with a range of 9%–18%. Their average upper bound was 27% over a range of 19%–38%. State and local income, property, and sales taxes add 3%–10%.



Government funding proved essential to these home-run labs on which I was fortunate to participate with colleagues and generations of graduate students. Our successes, combined with similar achievements by others at my university and elsewhere, indicate the critical importance of continued research investment for our nation's prosperity and for strengthening our national defense. It is the responsibility of those of us whose careers benefited from GoFAR to try to preserve the GoFAR opportunity for future generations of researchers.

America since its founding has capitalized on innovation as the engine of wealth creation and global prosperity. History is full of nations that failed to innovate and faded. Neither Ancient Rome nor Medieval China started an industrial revolution. Despite having brilliant scientists and mathematicians, the Soviet Union fell behind on technology and lost the Cold War. Heeding history, while China's GDP tripled since 2010, it steadily increased the fraction of its GDP invested in research and development every year since then [35]. If instead, the US forgets the past and decimates GoFAR, it could follow in the footsteps of the former "Cities on the Hill" who gave up their place as beacons of talent and innovation and whose positions in the world ultimately fell.

## Acknowledgements

A paper like this makes one reflect on their academic careers. As proud as I am of our research accomplishments, it's the personal relationships that one treasures. Thanks go to the hundreds of students and postdocs—too numerous to list—who worked hard to realize the visions of these labs, many of whom have made large marks of their own. The labs also relied on dozens of long serving staff, including Kattt Atchley, Damon Hinson, Roxanne Infanti, Tami Johnson, Jon Kuroda, Ken Lutz, Bob Miller, and Terry Lessard-Smith. And special thanks to the dozens of faculty colleagues whom I was honored to work with over four decades: Tom Anderson, Krste Asanović, Eric Brewer, Bob Brodersen, David Culler, Jim Demmel, Al Despain, Richard Fateman, Armando Fox, Mike Franklin, Joey Gonzalez, Joe Hellerstein, Paul Hilfinger, Dave Hodges, Mike Jordan, Anthony Joseph, William Kahan, Dick Karp, Randy Katz, John Kubiatawicz, Kurt Keutzer, Bora Nikolić, John Ousterhout, Raluca Popa, Jan Rabaey, Koushik Sen, Carlo Séquin, Scott Shenker, Ion Stoica, Mike Stonebraker, John Wawrzynek, Kathy Yelick, and Matei Zaharia.

I also thank Jeff Dean, Mark Hill, Jiantao Jiao, Ed Lazowska, Jasper Rine, Ray Simar, K. Tighe, Laura Waller, and Cliff Young who, in addition to some of my collaborators above, gave feedback that improved this paper.

---

[SIDEBAR TO PLACE NEAR START OF THE PAPER]

## Encouraging Impact by Measuring It

When I arrived as a new assistant professor in 1976, I was told by my university that they valued impact; don't fixate on paper count, research dollars, or solo authorship. I loved that liberating advice, and it set me on what proved to be a successful path.

Impact is not as easy to measure as citation counts or the [H-Index](#), where bigger is better. That might be one reason why more universities don't explicitly rely on impact for tenure and promotion cases, like what I was told. Alas, traditional scholarly measures don't always faithfully reflect impact. For example, the most popular paper from the plausibly most impactful project in Table 1 has only 5% of the citations of the highest one.

To better document this impact and encourage high-impact research, I'm working with Laude Institute to develop an "Impact Index." Inputs include professional society awards, open source project popularity, successful startups, corporate recognition, and so on (see Table 1). While the research-to-award gap can be long, open source popularity and startup success can be evident even faster than citation accumulation so they can help early in careers. Another benefit for more junior researchers is the Impact Index can highlight high impact role models for them to emulate.

---

## References

1. Anderson, Michael, 1990. [Strength \(and safety\) in numbers](#), *Byte Magazine* 15(12), pp. 337-339.
2. Anderson, Thomas, David Culler, and David Patterson, 1995. [A case for NOW \(Networks Of Workstations\)](#). *IEEE Micro*, 15(1), pp. 54-64.
3. Armbrust, Michael, Armando Fox, Rean Griffith, Anthony Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia, 2010. [A view of cloud computing](#). *Communications of the ACM*, 53(4), pp. 50-58.
4. Asanović, Krste, Ras Bodik, Bryan Catanzaro, Joseph Gebis, Parry Husbands, Kurt Keutzer, David Patterson, William Plishker, John Shalf, Samuel Williams, and Kathy Yelick, 2006. [The landscape of parallel computing research: A view from Berkeley](#), EECS Department, UC Berkeley Technical Report No. UCB/EECS-2006-183.
5. Asanović, Krste, Rimas Avizienis, Jonathan Bachrach, Scott Beamer, David Biancolin, Christopher Celio, Henry Cook, Daniel Dabbelt, John Hauser, Adam Izraelevitz, Sagar Karandikar, Ben Keller, Donggyu Kim, and John Koenig, 2016. [The Rocket chip generator](#), EECS Department, UC Berkeley Technical Report No. UCB/EECS-2016-17.
6. Asanović, Krste, and David Patterson. [The case for open instruction sets](#). *Microprocessor Report*, August 18 2014, pp. 22-24.
7. Boden, Nanette, Danny Cohen, Robert Felderman, Alan Kulawik, Charles Seitz, Jakov Seizovic, and Wen-King Su, 1995. [Myrinet: A gigabit-per-second local area network](#). *IEEE Micro* 15(1), pp. 29-36.

8. Brooks, Fred, and Ivan Sutherland, [Evolving the high performance computing and communications initiative to support the nation's information infrastructure](#), National Academy Press, Washington, 1995.
9. Bush, Vannevar. [Science—the endless frontier: a report to the president on a program for postwar scientific research](#). Washington, D.C.: National Science Foundation, 1945.
10. Catanzaro, Bryan, Narayanan Sundaram, and Kurt Keutzer, 2008. [Fast support vector machine training and classification on graphics processors](#). 25th International Conference on Machine learning, pp. 104-111.
11. Catanzaro, Bryan, Shoaib Kamil, Yunsup Lee, Krste Asanovic, James Demmel, Kurt Keutzer, John Shalf, Kathy Yelick, and Armando Fox, 2009. [SEJITS: Getting productivity and performance with selective embedded JIT specialization](#). *Programming Models for Emerging Architectures* 1(1), pp. 1-9.
12. Christensen, Clayton, 1997. [The innovator's dilemma: when new technologies cause great firms to fail](#). Boston, MA. Harvard Business Review Press.
13. Clark, Douglas, and William Strecker, 1980. [Comments on "The case for the reduced instruction set computer," by Patterson and Ditzel](#). *ACM SIGARCH Computer Architecture News* 8(6), pp. 34-38.
14. CBO (Congressional Budget Office), January 2025. [The Budget and Economic Outlook: 2025 to 2035](#).
15. Cummings, Jonathon and Sara Kiesler, 2005. [Collaborative research across disciplinary and organizational boundaries](#). *Social Studies of Science* 35(5), pp. 703–722.
16. Hennessy, John, David Patterson, and Christos Kozyrakis. *Computer architecture: a quantitative approach*. 7th edition, Elsevier, 2026.
17. Hill, Mark, 2016. [The “tire tracks” diagram corrected and humanized by National Academy workshop report](#) CCC (Computing Community Consortium) blog.
18. Kehe, Jason, March 25, 2025, [Angelina Jolie was right about computers](#), *Wired Magazine*.
19. Lee, Peter, 2012. [Continuing innovation in information technology](#) National Academy Press, Washington.
20. Lenoski, Daniel, James Laudon, Kourosh Gharachorloo, W-D. Weber, Anoop Gupta, John Hennessy, Mark Horowitz, and Monica Lam, 1992. [The Stanford DASH multiprocessor](#). *IEEE Computer* 25(3), pp. 63-79.
21. Massiglia, Paul, 1994. [The RAID Book: A Source Book for RAID Technology](#)
22. Mynatt, Elizabeth, 2020. [Information technology innovation: resurgence, confluence, and continuing impact](#) The National Academies Press. <https://doi.org/10.17226/25961>.
23. Nishimoto, Shinji, An Vu, Thomas Naselaris, Yuval Benjamini, Bin Yu, Jack Gallant, 2011. [Reconstructing visual experiences from brain activity evoked by natural movies](#), *Current Biology*.
24. Patterson, David, 1985. [Reduced instruction set computers](#). *Communications of the ACM*, 28(1), pp. 8-21.
25. Patterson, David, 2009. [Your students are your legacy](#). *Communications of the ACM*, 52(3), pp. 30-33.
26. Patterson, David, 2014. [How to build a bad research center](#). *Communications of the ACM*, 57(3), pp. 33-36.
27. Patterson, David, 2022. [2022 Draper Prize Acceptance Remarks](#).
28. Patterson, David, and David Ditzel, 1980. [The case for the reduced instruction set computer](#). *ACM SIGARCH Computer Architecture News*, 8(6), pp. 25-33.
29. Patterson, David, Garth Gibson, and Randy Katz 1988. [A case for redundant arrays of inexpensive disks \(RAID\)](#). ACM SIGMOD International Conference on Management of Data, pp. 109-116.
30. Rosenblum, Mendel, and John Ousterhout, 1992. [The design and implementation of a log-structured file system](#). *ACM Transactions on Computer Systems* 10(1), pp. 26-52.
31. Rowen, Christopher, Steve Przybylski, Norman Jouppi, Thomas Gross, John Shott, and John Hennessy, 1984. [A pipelined 32b NMOS microprocessor](#). IEEE International Solid-State Circuits Conference. Digest of Technical Papers, vol. 27, pp. 180-181.
32. Sherburne, Robert, Manolis Katevenis, David Patterson, and Carlo Séquin, 1984. [A 32b NMOS microprocessor with a large register file](#). IEEE International Solid-State Circuits Conference. Digest of Technical Papers, Vol. 27, pp. 168-169.
33. Smythe, Ian and Ian Ferguson, August 18 2014. The Case for Licensed Instruction Sets. *Microprocessor Report*, pp. 25-27.
34. Stonebraker, Michael, and Lawrence Rowe, 1986.. [The design of POSTGRES](#). *ACM Sigmod Record*, 15(2), pp. 340-355.
35. Textor, C, 2025. [Share of GDP expenditure on research and development \(R&D\) in China from 2010 to 2024](#), Statistica,
36. Wang, Randolph, and Thomas Anderson, 1993. [xFS: A wide area mass storage file system](#). 4th Workshop on Workstation Operating Systems, pp. 71-78.
37. Waterman, Andrew, Yunsup Lee, David Patterson, and Krste Asanović, 2011. [The RISC-V instruction set manual. Volume I: Base user-level ISA](#). EECS Department, UC Berkeley, Tech. Rep. UCB/EECS-2011-62 116, pp. 1-32.
38. Williams, Samuel, Andrew Waterman, and David Patterson, 2009. [Roofline: an insightful visual performance model for multicore architectures](#), *Communications of the ACM* 52(4), pp. 65-76.
39. Zaharia, Matei, Mosharaf Chowdhury, Michael Franklin, Scott Shenker, and Ion Stoica, 2010. [Spark: Cluster computing with working sets](#). 2nd USENIX workshop on Hot Topics in Cloud Computing (HotCloud 10).