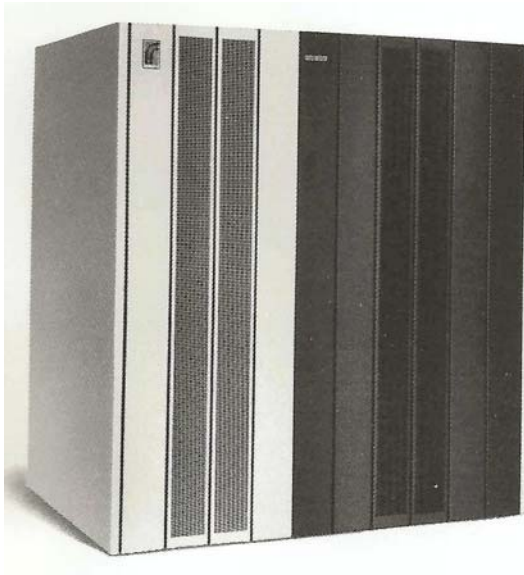


SAMPLE PAGES FROM MULTIFLOW COMPUTER: A START-UP ODYSSEY

Multiflow Computer

A Start-up Odyssey



Elizabeth Fisher

CONTENTS

Preface	1
The Eureka Moment	7
Yale	31
VLIW	49
John and John	63
Three Guys and a Car	75
Multiflow Computer	90
Don Eckdahl	109
The Engineering Team	125
Building the Trace	139
Product Introduction	151
No Vectors	167
Le Flic	181
Selling the Trace	196
Doldrums	210
Europe	222
The End	244
Epilogue	258
Technical Appendix Written by Josh Fisher	268
Sources and Further Reading	278
Acknowledgements	284

PREFACE

This is a story I have always wanted told. For years I begged Josh to write it, but he is a scientist and this story is not about science or, really, even about technology. Gene Pettinelli, the Multiflow lead investor, asked Josh to give talks about the company but he always refused. He never understood what we found so fascinating, never understood the appeal. Bob Rau, Josh's colleague at Hewlett-Packard who founded a Multiflow competitor, always said that a start-up was a fabulous thing to have done—once. And that was Josh's attitude: it was an amazing experience but exhausting, and only the technology interested him.

Josh is my husband, Joseph A. Fisher: computer scientist, retired Hewlett-Packard Senior Fellow, former Yale professor, Eckert-Mauchly award winner. He invented a new computer architecture in his PhD thesis at NYU and developed it as a young faculty member at Yale. He thought his radically different approach to computer design would revolutionize the world of scientific computing, and he came nearer to realizing that dream than seems possible.

Josh put his heart into Multiflow, even though the role he took wasn't doing science and he had to venture into management. During these years, I watched him change with blinding speed from a scruffy graduate student with working-class roots to a respected scientist and seasoned senior executive. It was a truly remarkable transformation.

Multiflow Computer, Inc. was a computer start-up, founded in 1984 to commercialize Josh's VLIW technology. It was funded by venture capital, founded by Josh and by John O'Donnell and John Ruttenberg, both also from Yale. The company manufactured computers, selling and distributing them nationally and internationally, until it went out of business six years after its founding. It raised about \$60 million of venture capital, sold about 140 machines at an average price of around \$350,000 and had about 170 employees at its end. One of the computers, a Multiflow Trace, now lives in the Computer Museum in Mountain View, California.

When Josh first publicly proposed VLIW technology in 1983, his scientific colleagues thought it wouldn't work, that you couldn't build a general purpose computer to run any scientific program with speed-up coming from the software not the hardware. This radically different computer design was too strange for many of his colleagues; everyone was skeptical.

Jim Goodman, Chair of Computer Science at the University of Wisconsin, saw the first VLIW presentation in Stockholm. "When I first heard it, I thought it was the most 'blue sky' idea ever, and the next thing I knew, there was a working product." Jim said it was the most amazing thing he had ever seen.

What made Josh and his colleagues think they could completely change the computer industry? They were young, of course, and had the arrogance of youth, but the 1980s was also a time for grandiose ideas in the computer world. Computer science was a new science then and the rules were just being written. The way computers were used was changing quickly; companies with new technologies were popping up every day. It was the Wild

West, fun, with new discoveries everywhere. It seemed that anything could happen.

Josh and his colleagues had no idea what they were getting into when they started Multiflow. They learned as they went along, making their way in the industrial world, unfamiliar as it was to most of them. Multiflow had timing on its side, and a lot of luck, and whether any good idea can survive to change the world depends as much on luck as on the superiority of the idea. Ultimately, though, they never sold enough computers to be profitable and the administration and sales efforts weren't up to the stellar quality of the engineering. Then the changing nature of the computer industry, which helped at first, swept over Multiflow and drove it and its whole industry segment under. It was "the attack of the killer micros," the beginning of the technological convergence that continues today, that made the computers Multiflow produced—powerful mini-supercomputers—unnecessary for most of scientific computing.

The work was hard, a roller coaster of emotional highs and lows—the travel, the craziness, raising money, the selling—but if it had succeeded, Josh never would have left. For him it was very personal and, really, a dream—all those amazingly talented people at Multiflow working on his technology, on his vision. He wanted to change the world of scientific computing and they were making it happen.

The story of Multiflow is also about the remarkable people who worked there, the people who went on to become stars in the computer industry. This was a group of very smart and talented young engineers—first rate computer scientists every one of them—full of drive and strong opinions, believing in the technology and creating the product. They worked hard, working with people who were at the same genius level of intelligence as they were,

people who would be stars anywhere—with management completely behind them. It was paradise for these smartest of engineers.

Besides Josh, the company produced another Eckert-Mauchly award winner and three other industrial Fellows. Even many of the non-engineers were outstanding. Mike Loukides, a Stanford English PhD who wrote the documentation, went on to become a Senior Editor at O'Reilly Media, a prominent publisher of technical books, after Multiflow folded. He is now their Vice President of Content Strategy.

I had a peripheral role at the company, first finding office space and employee benefits and then producing the presentation graphics used in fundraising and sales. But mostly my role was to be Josh's wife, listening, amazed at the scale of what was happening. This is how I heard the whole Multiflow story, from Josh at our dinner table and in phone calls from every part of the world when he was on the road. I thought Multiflow was the most exciting thing I had ever been a part of. Josh and I have been telling each other these stories for the past twenty-five years.

People have asked me whether Multiflow made my family rich and the answer is no, or not directly. Josh invented the technology but because it is an architectural style, he doesn't own it, any more than Little Richard owns Rock and Roll because he invented it. Little Richard is rich because he plays Rock and Roll so well, but anyone can play music in that style without paying royalties to Little Richard.

Many times along this road we thought we might be very rich; there was certainly potential for great wealth. We owned unregistered stock worth \$40 million at one point, valued at the price investors actually paid. Then when Multiflow went under, our stock was worthless since all the

company assets, including technology licenses, went to pay its debts. There was also a question then of whether Multiflow's creditors would go after our personal assets, impoverishing us, even though Josh had never personally guaranteed the debt. None of that happened, though; we didn't get rich, but we didn't end up in financial ruin.

One person did become rich because of Multiflow, but it was someone who never worked there. Steve Wood had been a graduate student at Yale with many of the first Multiflow employees but had left Yale in 1983 to join a software start-up in the northwest. When Multiflow began, all his friends wanted Steve to work with them. They told him what fun they'd have working on Josh's technology, developing a product that would change the world. His friends had no respect for what Steve was doing—not only software, but personal computing software that was nowhere near the glamorous scientific computing world. They thought his job was beneath him and urged him to join them, doing something really important.

Multiflow made Steve an offer, a better offer than to any of the other engineers: \$45,000 salary plus relocation benefits and 22,000 shares of stock. Steve was sorely tempted but he really liked his job. His boss wanted him to stay and offered him a large amount of unregistered stock as an inducement. His boss said that someday this company, Microsoft, would go public and Steve's stock would be worth something. Steve stayed at Microsoft—and it happened just as Bill Gates said it would.

We didn't get rich, but Multiflow made Josh's scientific reputation and also his reputation for recognizing talent and putting together fabulous technical teams. It also made the reputations of virtually all the engineers who worked there. There was a guy who started work the week

Multiflow went under who benefited from its aura—he got a job right away, burnished by being chosen by Multiflow.

VLIW technology also succeeded, even though Multiflow failed as a company. No longer an unknown technology, VLIW now gets nearly a million Google hits at last count. It did not change all scientific computers as Josh and his colleagues thought it should but, through Multiflow licensing agreements and independent development efforts, it has spread throughout the technical community. In 2000, Josh's group at HP Labs developed a VLIW processor with STMicroelectronics, one of the world's largest manufacturers of semiconductors. As of 2012, ST has sold chips with over 500 million of these processors to manufacturers for use in digital video, and HP uses the processor extensively in its printers. In addition, VLIW technology is now used industry-wide as embedded processors in printers, smartphones, computer graphics chips, cable boxes, cell base stations, GPS devices and DVRs. As a result, there is something with VLIW technology in almost every home in the developed world.

THE EUREKA MOMENT

On the Wednesday before Thanksgiving, 1978, Josh was working on his doctoral thesis and waiting for me to come home from my office in New York City. He was in the public library near our home in Teaneck, New Jersey, and when I got home, he, I and our eleven-month-old son, Davey, would drive to Boston to spend the holiday with my brother and sister-in-law.

Josh was a graduate student in computer science at NYU's Courant Institute, and for several years he had worked on a faculty project to emulate the department's supercomputer, a CDC 6600, creating a handmade computer called PUMA. Now he was working on the hardest problem he found on this project, one he hoped to turn into his thesis.

Josh had been worrying at this problem for months and was just starting to see it clearly. Because I was late getting home that Wednesday before Thanksgiving, Josh had extra time in the Teaneck library to work on his thesis, extra time to think deeply, to get really involved—to have his eureka moment.

* * *

In 1978, Josh was 32, older than most computer science graduate students. His life had been less orderly than theirs, and his family background and personal history

were also different. Unlike him, many of his fellow students came from upper middle class backgrounds and progressed from college to graduate school with a firm eye on how they would spend their careers. Because Josh was different, his worldview was too. He had an outsider's perspective that shaped his approach to science, helping him to look at the world in unique ways.

His parents grew up poor in New York City, each the oldest of large families. Harry, his father, was not quite seventeen when Josh was born, six months after his parents were married; Sue, his mother, was eighteen: they had only \$5 between them when Josh was born. Sue dropped out of high school, but Harry graduated from the High School of Aeronautical Trades, his young family living with his parents in the Bronx while he finished high school. Harry wanted a technical career but, because he had a



**Josh as an infant with his teenaged parents,
Harry and Sue**

family to support, he took a job at Howard Clothes, a men's clothing chain, after he graduated from high school.

*END OF PREVIEW PAGES FOR THIS CHAPTER
MORE PAGES BELOW*

YALE

FIRST PAGES REMOVED FROM THIS CHAPTER

Josh's computer architecture class, a joint graduate/undergraduate course, became the exciting systems class to take in the Department. Everyone wanted in. And many students who took it became interested in trace scheduling. With Alan not leading any research, Josh ran the only practical systems project in the department. It was a natural for some of the students, especially ideal for a core group of smart systems graduate students who were used to working together. They weren't all part of Josh's project, but they all followed its progress.

The Tools Group, as they called themselves, was a dynamic, fun-loving bunch of students who delighted in building software for the whole department to use. They were the smartest of the smart, and they designed tools for practical use in a spirit of altruism. They built text editors, email programs, and graphics interfaces for the departmental computers, the terminals and the Imagen laser printer, working with macho exuberance because the work was so exciting, so much fun and because they could. If a system couldn't do something, they built a tool to make it do that something, working separately or together, each

one writing a module. They, too, had learned from Perlis about building as research, and they delighted in it. It was a great laboratory for these smart, creative people who could reason so imaginatively about computers.

The Tools Group thought Steve Wood was their best builder, but Josh thought it was John Ellis and asked him to be his teaching assistant. The other graduate students were stellar builders, too: Bob Nix, Doug Baldwin, Geoff Lowney, Nat Mishkin, John Ruttenberg, Richard Kelsey, as were undergraduates, Steve Wager, Olin Shivers, Tom Karzes, and Ben Cutler.

Before the internet, before better graphics, before widespread access to powerful computing and applications to do anything you want, long before hacking meant criminal activity, the students delighted in their abilities to make the computer perform. The group played tricks on each other, the undergraduates getting into battles to hack into each other's accounts and the departmental machines. At one point all the error messages from the Imagen printer started "Swine!..."

Calling someone a hacker was a badge of honor in those days, a sign your peers respected you. The best hackers were the most accomplished, the ones everyone looked up to—what everyone wanted to be. None of the Yale students thought they had anything but the purest intent, and they reveled in what they could do.

One morning Josh turned on his terminal to find it suddenly going blazingly fast, much faster than it had gone the day before. A couple of the graduate students had gone into the computer system the night before, adjusted some of the settings and added some routines here and there to speed up the communication time with the terminals. The Systems Administrator, proprietary about the computers, had steam coming out of his ears. He talked about

disciplinary action and soon slowed the machine back down. Then a memo appeared, purporting to be from him, describing the horrors of having the machine go fast. It was ironic and gentle but to the point, beautifully written, since the actual authors were Yale students. Ultimately, though, the lure of a faster computer was too much, and the speed-up put in by the Tools Group was restored. A few years later, John O'Donnell became Systems Administrator and had none of these problems. John was in there with the hackers, changing things and fiddling with the systems, making everything go faster.

*END OF PREVIEW PAGES FOR THIS CHAPTER
MORE PAGES BELOW*

VLIW

In the fall of 1982 Josh began talking publicly about hardware, about a new style of computer, designed to run trace scheduled code. He called this style VLIW, Very Long Instruction Word technology. To officially launch it, he submitted a paper to the International Symposium of Computer Architecture, the most influential architecture forum, for a conference to be held the following June in Stockholm.

To help him with the ISCA paper, he hired Mary-Claire Van Leunen, a departmental grant writer whose recently published book, *A Handbook for Scholars*, was a bible in the Yale Computer Science Department. Famously prickly and outspoken, she held a going-away party for one of the departing faculty, a celebration—after he left. Mary-Claire was an honorary member of the Tools Group.

It took Josh a while to warm to Mary-Claire because she had many strict rules for academic papers. She thought they were mostly poorly written by the scientific community—and she was adamant. At first Josh thought her expertise was limited to “fourteen rules about commas” and reference citations, but when she told him that academic writing was like writing programs, that the goal was simplicity, transparency and clarity, he realized that she was someone who knew a lot.

“Only one idea per paragraph,” said Mary-Claire. “As simple as possible.”

She encouraged Josh to name his computer the ELI-512, Enormously Long Instructions, saying everyone would know that it “wasn’t invented at Harvard,” since Yalies were called Elis after Elihu Yale. In the computer name, 512 was the number of bits in each instruction word.



Article on the ELI-512 in the Yale Daily News, April, 1983

With Mary-Claire’s help and months of work, Josh’s ISCA paper was beautifully written: simple and clear. Because of her lessons, his reviews said, “This is such a clearly written paper it should be published on those grounds alone,” but also, “This paper was written in an annoyingly juvenile style,” which Josh took as a compliment.

END OF PREVIEW PAGES FOR THIS CHAPTER
 MORE PAGES BELOW

THREE GUYS AND A CAR

Before Josh, Ruttenberg, and O'Donnell got too far along with the venture capitalists, something new popped up and they found themselves going in a different direction.

Apollo Computer was a workstation company O'Donnell brought to Yale to supply departmental computers. It was founded in 1980 and had recently gone public, making the founders millionaires many times over. Josh and O'Donnell hadn't approached Apollo the summer before to build the VLIW machine because they thought it was too small with too many growing pains to dedicate a team to a new technology, even with the advice of the Yale group. But O'Donnell had been encouraging the founders to invest in the company he was starting.

Apollo wanted to get into the market for higher performance computers and saw a lot of potential in Josh's technology. In October, 1983, Josh, John and John drove to Apollo headquarters in Chelmsford, MA to meet with Mike Greata, a founder and vice president of engineering in charge of new ventures, to see if there was any business they could do together. They met all day, Greata looking at the business plan they had put together for the VCs. He liked what he saw and had an innovative idea.

Greata suggested that instead of investing, Apollo might want to entirely fund the new company, cutting out the venture capitalists completely. The new company would design the computer and Apollo would manufacture and market it as its own product. Eventually, Apollo would buy them out. This would be good for both groups, Greata told them. The Yale group would have funds to develop the computer and Apollo would have a new product, giving them entrée into the high performance market.

This was a startling development—something new to think about. They told Mike they would consider it seriously and went home to get their minds around the new idea.

The three of them had focused on VC financing since the meeting with Fairfield Ventures the month before, but this seemed even better—much less risky. With funding from Apollo, the new company could do what Josh's group had always been good at—designing and building technology—and leave manufacturing, marketing and sales to people who knew how to do them. They might not get as rich, but there was a much smaller down-side risk, since funding would be assured. Even Ruttenberg, such a proponent of venture capitalism, thought a deal with Apollo was preferable to going it alone.

*END OF PREVIEW PAGES FOR THIS CHAPTER
MORE PAGES BELOW*

MULTIFLOW COMPUTER

On April 3, 1984, less than two weeks after the founders met George Thibeault, Multiflow Computer was incorporated in Delaware, home to most of the Fortune 500 companies. The next week they borrowed \$500,000 from Apollo and a week after that, Josh requested a two year leave of absence from Yale and ordered business cards for himself as President. He wanted to add CEO, but Ruttenberg told him that such grandeur was unseemly for a company as small as Multiflow.

They hired three employees who had accepted job offers; Bob Nix was the first employee to get a paycheck. Geoff Lowney joined, then Tom Karzes, Ben Cutler, and Nat Mishkin—all former Yale students, and they expected a lot more to join. New people started every week, sometimes more than one a week. The founders recruited everyone, interviewing friends, and friends of friends, and friends of friends of friends. When Josh tried to steal an administrator from Yale, Roger Shank told him that Yale was not his Little League.

“Not your ‘farm team,’ you mean,” Josh corrected, and Roger became the first of many bosses to say that Josh was the most frustrating person he ever had working for him.

PAGES REMOVED



Leigh Cagan - Who ya gonna call?

After a while Leigh got sick of saying “Multiflow employees” and looked for something more descriptive. Finally, he came up with “Multifloids” and the name stuck. Somehow, the “oid” sounded computer-ish to him, if a little dorky. The engineers took it up and so did the founders; they all called themselves Multifloids.

In August, Multiflow moved from Chapel Street in downtown New Haven to Branford, CT on the coast, to the converted telephone company warehouse. Chapel Street had become very crowded and now there would be offices to work in, closed offices. The new space even had a loading dock. They settled in, going full speed to develop the VLIW computer.

And the newly minted Multifloids threw themselves into their work: Nat and Bob Nix took over the operating system under O’Donnell’s management, and Ruttenberg took over the compiler. Josh provided technical leadership,

but he was too busy recruiting and getting the business off the ground to be involved on a day-to-day basis. He worked with the engineers when necessary, but mostly he left them alone. He respected them and had faith in their abilities. They were so smart that Josh let their creativity dictate how the machine would progress. Although the technology was based on his work, he was confident that their efforts would complement his.

The group thrived with this much freedom, everyone wanting to produce the best results, sharing the joy of scientific collaboration. They were working hard, working together, with people who were at the same genius level intelligence as they were, people who would be stars anywhere, with management completely behind them. It was fun and they trusted each other. There was the thrill of the hunt, of going unexplored places in architecture, the feeling of untrammelled space. They were getting to do what they wanted, build what they wanted, be part of a project they knew would make history. It was paradise for these smartest of engineers.

Some of the new recruits were shocked when Josh couldn't be as closely involved with the technology as they wanted, but since the engineering work was exciting, and the test results were starting to come in strong, they quickly adapted to the leadership structure. The engineers meshed, the new ones and those who had been part of the Tools Group at Yale. They became an unstoppable team.

It was easy to find the first technical recruits, software people from Yale and NYU, but it was harder to find hardware people since the founders' expertise and everyone they knew was in software. And New Haven was not Boston; it was "Silicon Nowhere," as Apollo and the VCs said, a place hard to entice people to move to. They made one hiring mistake early on, hiring a hardware

engineer who was not up to the job and who ended up quitting. At that point, they went farther afield to the last reaches of their friendship networks looking for people to interview. And sometimes they found themselves talking to people very different from themselves.

*END OF PREVIEW PAGES FOR THIS CHAPTER
MORE PAGES BELOW*

THE ENGINEERING TEAM

FIRST PAGES REMOVED FROM THIS CHAPTER

Early on, the founders and Don had decided that the new computer would be called the Trace since trace scheduling was its driving principle. The machine they were building was inspired by the ELI-512 from Yale, but it was an entirely new machine, built completely from scratch. And they didn't want to use a name with Eli in it because it sounded regional, evoking Yale.

John O'Donnell had been working with VLIW concepts longer than any of the hardware engineers, so initially he was the one with the best overview of the computer they were building. Papworth and Rodman had built machines at Prime, but no one else in hardware had industrial experience. And many of the engineers were just out of school and had never done anything like this before, created anything as ambitious as a new machine. They were feeling their way along—all of them.

During Multiflow's first year O'Donnell, Papworth, and Rodman made all the high level Trace hardware design decisions—the backplane, the major data paths, the memory system, the floating point unit, and the integer

unit. These were crucial decisions, the backplane most of all, because it was the skeleton of the system, complicated by a lot of boards, connections and pins. How it was set up determined how signals would travel from board to board—and getting it right was crucially important because once it was manufactured, it would be too expensive to change.

And while the hardware people were designing their part, the compiler people began writing the program that would use trace scheduling to tell the hardware which instructions to execute together. The first three: Ruttenberg, Geoff Lowney and Tom Karzes, had all been at Yale, and they modeled the Multiflow Compiler after Yale's Bulldog Compiler. They used the computer language C, but when it couldn't handle everything they wanted it to do, they enhanced the language, almost as if they were creating an entirely new one, adding structures the programmers could use—precise, elegant embellishments giving them more power to do what they wanted.

Then, writing the compiler, each person handled something different—Tom the optimizer, Ruttenberg the back end and Geoff the trace scheduler. This is how large



**Geoff Lowney wrote the
trace scheduler**

programs are usually created, and at Multiflow it was a good thing because each of the guys had radically different work styles—and very intense personalities. Ruttenberg could only plan out a program while he was writing and would correct his mistakes as he went. Tom, on the other hand, needed long periods of planning, making everyone edgy, but then would quickly write up flawless programs. Tom could be intolerant of mistakes in other people's code, and mistakes were the natural byproduct of Ruttenberg's style of working. It was all the worse since Tom was so smart that he was usually right. Geoff was caught in the middle, reading the detailed public critiques Tom produced whenever he saw anything less than perfect—Geoff cringing as he got to his email every morning.

After a year Stefan Freudenberger joined the compiler group, a new PhD Geoff had recruited from NYU, and he added his focused relentlessness and methodical German precision to the mix. Stefan didn't work well with Tom's abrasiveness, even worse when you added Ruttenberg. He was so straight-laced that he told mild-mannered Geoff Lowney that he had to stop cursing or he couldn't work with him. And then Stefan had to work with the hardware guys—all strong personalities, too, and not given to moderate language.

END OF PREVIEW PAGES