

*The Hardy Boys and the Microkids  
build a computer*

## FLYING UPSIDE DOWN

BY TRACY KIDDER

ONE HOLIDAY MORNING IN 1978, TOM WEST TRAVELED to a city that was situated, he would later say guardedly, "somewhere in America." He entered a building as though he belonged there, strolled down a hallway, and let himself quietly into a windowless room. Just inside the door, he stopped.

The floor was torn up; a shallow trench filled with fat power cables traversed it. Along the far wall, at the end of the trench, enclosed in three large, cream-colored steel cabinets, stood a VAX 11/780, the most important of a new class of computers called "32-bit superminis." To West's surprise, one of the cabinets was open and a man with tools was standing in front of it. A technician, still installing the machine, West figured.

Although West's designs weren't illegal, they were sly, and he had no intention of embarrassing the friend who had told him he could visit this room. If the technician had asked West to identify himself, West wouldn't have lied and he wouldn't have answered the question, either. But the moment went by. The technician didn't inquire. West stood around and watched him work, and in a little while the technician packed up his tools and left.

Then West closed the door and walked back across the room to the computer, which was now all but fully assembled. He began to take it apart.

West was the leader of a team of computer engineers at a company called Data General. The machine that he was disassembling was produced by a rival firm, Digital Equipment Corporation, or DEC. A VAX and a modest amount of adjunctive equipment sold for something like \$200,000, and as West liked to say, DEC was beginning to sell VAXes "like jellybeans." West had traveled to this room to find out for himself just how good this computer was, compared with the one that his team was building.

West spent the morning removing the VAX's twenty-seven printed circuit boards. He'd take one out, study it, make a few notes, and then put it back. These boards were flat plates, each about the size of a shirt cardboard. In regular columns across their surfaces lay small rectangular boxes. Each of these boxes enclosed an integrated circuit, or "chip"; if bared and examined under a microscope, the chips would look like mazes—imagine

the wiring diagram of an office building inscribed on a fingernail. It's possible to get inside the chips, inside the littlest boxes inside the boxes that constitute the central works of a modern computer, and, bringing back the details, to create a functionally equivalent copy of a machine. "Reverse engineering" is the name for that art, and it takes time and equipment. West called such engineering "knock-off copy work." He had a simpler purpose. He was not going to imitate VAX; he just wanted to size it up.

Looking into the VAX, West felt that he saw a diagram of DEC's corporate organization. He found the VAX "too complicated." He did not like, for instance, the system by which various parts of the machine communicated with each other; for his taste, there was too much protocol involved. The machine expressed DEC's cautious, bureaucratic style. West was pleased with this idea.

His hands in the machine, West was also studying and counting parts; many of the chips had numbers on their housings that were like names familiar to him. When he was all done, he added everything up and decided that it probably cost \$22,500 to manufacture the essential hardware of a VAX. He left the machine exactly as he had found it.

"I'd been living in fear of VAX for a year," West said one evening afterward, while driving along Route 495 in central Massachusetts. "I wasn't really into G-2. VAX was in the public domain, and I wanted to see how bad the damage was. I think I got a high when I looked at it and saw how complex and expensive it was. It made me feel good about some of the decisions we've made."

West was forty but looked younger. He was thin and had a long narrow face and a mane of brown hair that spilled over the back of his collar. These days he went to work in freshly laundered blue jeans or pressed khakis, in leather moccasins, and in solid-colored long-sleeved shirts, with the sleeves rolled up in precise folds, like the pages of a letter, well above his bony elbows. He expostulated with his hands. When dismissing someone or some idea or both, he made a fist and then exploded it, fingers splaying wide. The gesture was well known to those engineers who worked for him. Long index fingers inserted under either side of the bridge of his glasses signified thought, and when accompanied by a long "Ummmmmmmmh" warned that some emphatic statement was near. Indeed, West made few statements that

were not emphatic. Seen at the wheel of his shiny red Saab, he made a picture of impatience. His jaw was set; he had a forward lean. Sometimes he briefly wore a mysterious smile. He was a man on a mission.

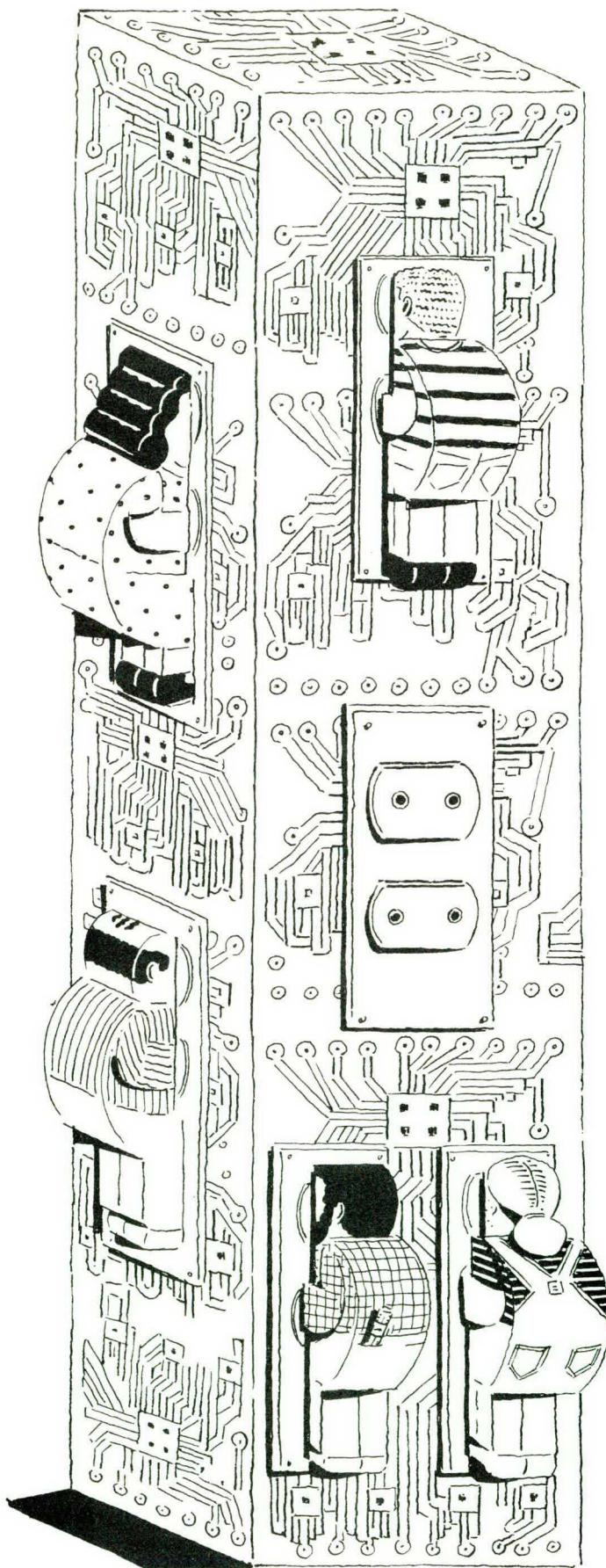
"With VAX, DEC was trying to minimize the risk," West said, as he swerved around another car. "We're trying to maximize the win . . ."

**I**N THE EARLY 1960s, SEVERAL COMPANIES BEGAN TO manufacture computers that were much less powerful but also much smaller and cheaper than the machines then in existence. These new devices were called minicomputers. By 1978, the increasingly imprecise term "minicomputer company" could be applied to about fifty corporations. Minicomputer sales had grown from about \$1.5 million worth of shipments in 1968 to about \$3.5 billion in 1978, and most interested parties believed that the business would continue to grow by about 30 percent a year.

DEC was one of the first minicomputer companies, and it was the largest corporation in this segment of the computer industry: the IBM of minis. In 1968, three young computer engineers who left DEC and a salesman from another company founded Data General. Minicomputer companies were known for playing rough; Data General had acquired a reputation as one of the roughest of them all. "The Darth Vader of the computer industry" was the way one trade journalist described the company. Meanwhile, Data General thrived. It made good computers inexpensively and it managed its business adroitly. By 1978, Data General was taking in about half a billion dollars a year. It was only ten years old, and its name had just been added to the list of the nation's 500 largest industrial corporations. Moreover, for most of its history Data General had maintained the highest profit margins in the computer industry, after IBM.

Making computers is a risky enterprise. Young, successful computer companies often get into serious trouble, largely because success in their business means rapid, stressful growth. Data General grew by more than 30 percent a year for a decade, and all the while the technology of computers was changing.

At some computer companies, it has fallen mainly to engineers, working below decks, as it were, to make the first decisions about new products. Data General was such a company, and one often heard that its president, Edson deCastro, himself a very successful computer engineer, liked "self-starters." By 1978, though the company's balance sheet had never looked better, it was becoming apparent that Data General had need of initiative from its engineers. Into the world of the minicomputer had come a new kind of machine—the 32-bit supermini. DEC's VAX was the best-known example of such a machine, and several other of Data General's rivals had also produced superminis. Data General, meanwhile, had



yet to offer one of its own. "A disaster," Tom West said of this situation.

The most important characteristic of the 32-bit mini was its system of storage. Storage in a computer resembles a telephone system, in the sense that every piece of information in storage is assigned a unique number, so that it can be readily found. If the standard length of a phone number is seven digits, then enough unique numbers can be generated to serve the needs of New York City; but if a three-digit area code is added, every telephone customer in America can have a unique number. The 32-bit supermini was a computer with an area code. Since the advent of Data General, most minis had been "16-bit" machines. The standard length of the numbers that such a machine assigns to items in its storage is 16 bits, 16 binary digits. A 16-bit machine can directly generate only about 65,000 unique numbers for its storage system. A 32-bit machine, however, can directly generate some 4.3 billion different numbers.

All interested parties agreed that the demand for superminis would be huge; the market might be worth several billion dollars by the 1980s, some said. If Data General failed to produce a 32-bit machine or something equivalent, it could expect to lose some old customers, and, perhaps more important, it would forfeit one of the next decade's best opportunities for gaining new business. The company could not now be the first to enter this new market, but that was all right; sometimes it was better not to be first. However, Data General had to field a suitable machine fairly soon, because customers get married to computer companies in intricate ways, and once they've married elsewhere they're often gone for good. Time was running out, Tom West maintained. "We're gonna get schmeared if we don't react to VAX," he said.



**D**ATA GENERAL'S HEADQUARTERS STAND NEAR THE intersection of two superhighways some thirty miles west of Boston. It is a low-lying brick building with TV cameras mounted on the corners of its roof, and all in all looks like a fort. Its official name is Building 14 A/B. Inside, it is essentially divided into an upstairs and a downstairs. The executives work upstairs. The lower level of Building 14, subterranean in front and at

ground level in back, is another country. It belongs mainly to engineers.

West led the way down into this region one evening in the late fall of 1978, through confusing corridors and past mysterious doors that were locked up and bearing signs that read "RESTRICTED AREA." Then the hallways ended, and all around, under fluorescent light, lay fields of cubicles without doors. Their walls stood too low for privacy. Most contained a desk with a computer terminal on it. In many, there was a green houseplant. Green plants poked their heads, like periscopes, above the cubicles' walls. "The great statement," said West, gesturing at the foliage and smiling faintly. "It's basically a cattle yard."

By day, the basement held a homogeneous-looking throng, made up largely of young white males wearing jeans and corduroys and hiking boots; few wore neckties, but neat grooming was the rule. Now and then a visitor might catch a glimpse of a fellow with wild hair, dressed in Army-surplus clothes, but such figures were rare.

West's team specialized in the design and development of the hardware of new computers. It was only one of several such teams at Data General, and it was not the largest or, in the fall of 1978, the most prestigious. It was named the Eclipse Group, after the current generation of 16-bit Data General computers. The Eclipse Group, which numbered about thirty then, occupied a portion of a field of cubicles and a few narrow, windowless offices, one of which belonged to West. No sign announced that this was the group's territory. At night, it did seem that more lamps burned on in the Eclipse Group's offices and cubicles than in many other parts of the basement. At some moments during the day, the area had the atmosphere of a commuter train, and at others it reminded one of a college library on the eve of exams: silent and intent youngsters leafing through thick documents and peering into the screens of computer terminals. Conversation, especially the speech of the senior engineers, contained words and phrases such as these: a *canard* was anything false, usually a wrongheaded notion entertained by some other engineering group or other company; things could be done in ways that created *no muss, no fuss*, that were *quick and dirty*, that were *clean*. *Fundamentals* were the source of all right thinking, and weighty sentences often began with the adverb *fundamentally*, while *realistically* prefaced many flights of fancy. There was talk of *wars, shootouts, hired guns*, and people who *shot from the hip*. *The win* was the object of all this sport, and *the big win* was something that could be achieved by *maximizing* the lesser one. From the vocabulary alone, one could have guessed that West had been there and that these engineers were up to something special.

In fact, they were building their own 32-bit supermini, a machine that West fervently hoped would be a worthy rival to DEC's VAX and maybe the basis for Data Gener-

al's ascent in the Fortune 500. Oddly, though, West and some of the senior engineers on his team expressed the paradoxical feeling that they were building a machine absolutely essential to the company but were doing it largely on their own. "I think we're doing it in spite of Data General," said one of West's lieutenants in the middle of the project.

Setting up intramural competition among various parts of a company is an old strategy of management. Many firms in the computer industry, most notably IBM, have used it; they deliberately establish internal competition, partly on the theory that it's a useful prelude to competition with other companies. At Data General, such internal struggle had the name "competition for resources." An engineering team such as the Eclipse Group sometimes had to vie with other engineering groups for the right to produce a new computer. A year or so before, some members of the Eclipse Group had found themselves in such a competition, against a much larger team of Data General engineers, situated in North Carolina. The Eclipse Group had been competing with the team in North Carolina essentially for the right to produce Data General's supermini, what West would later call "the answer to VAX." The Eclipse Group's project had been scrapped. There had been an intramural competition for resources, and the Eclipse Group had lost. But West had decided not to abide by the decision. He had launched the Eclipse Group on another big project, one that would rival North Carolina's. Doing so had taken him some months. It had also required that he pursue some indirect measures.

West had believed that whatever its other virtues, the machine that the company engineers in North Carolina were building did not represent a timely solution to the problem that DEC's VAX posed for Data General. West also wanted to save the Eclipse Group and himself from the fate of working only on small projects. So he had borrowed ideas from anyone who had some to share, and by the very early spring of 1978 he had settled on a new plan. The Eclipse Group would build a schizophrenic computer, one that would work as both a 16-bit Eclipse and a 32-bit supermini. The proposed machine was nicknamed "Eagle."

The production of software, the programs that tell computers what to do, costs customers time and money and sometimes entails awful administrative problems. Eagle would protect old customers' substantial investments in 16-bit Eclipse software and would offer prospective buyers at least the possibility of savings in software development. At the same time, this machine would fulfill Data General's need for a computer with enlarged "logical address space." And West thought that the front office was likely to let the Eclipse Group build this computer, if it was presented correctly.

"You gotta distinguish between the internal promotion to the actual workers and the promoting we did to other

parts of the company," West later explained. "Outside the group, I tried to low-key the thing. I tried to dull the impression that this was a competing project with North Carolina. I tried to sell it externally as not much of a threat. I was selling insurance; this would be there if something went wrong in North Carolina. It was just gonna be a fast, Eclipse-like machine. This was the only way it was gonna live. We had to get the resources quietly, without creating a big brouhaha."

And so, when he proposed the idea to people outside his group, West made Eagle appear to be a modest project, and he got permission to go ahead. But when he proselytized engineers who might help build this machine, it was clear that West's intentions weren't modest at all.

From the point of view of a purist ("technology bigot" is the usual term), Eagle in its vague outlines looked messy. Indeed, some engineers called the plan "a kludge," computer jargon for any ill-conceived thing. West varied his pitch to suit his audience. His general remarks ran as follows: Eagle might not look it from the outside, but in fact it was going to be a new, a fast, a "sexy" machine. It would be software-compatible with 16-bit Eclipses, not because it was going to be just another Eclipse, but because that feature would make it a "big win" commercially. They were going to build Eagle in record time, working "flat out by definition," because the company needed this machine desperately. And when they succeeded and Eagle went out the door with their names on it, as West put it, and started selling like jellybeans, then they would all be heroes.

Once in a while West and some members of his staff asked themselves whether the company's president, deCastro, might not have orchestrated everything, including their feeling that they were on their own. Whatever its origin, though, that feeling was evidently invigorating. "Anytime you do anything on the sly, it's always more interesting than if you do it up front," one of West's lieutenants remarked. West said, wearing his wry smile, "We're building what I thought we could get away with."

**B**Y THE SPRING OF 1978, WEST HAD GATHERED A CADRE of fairly experienced engineers. But to build Eagle, it was soon clear, more engineers were needed.

West conferred with an old colleague named Carl Alsing. Alsing was in his mid-thirties, a veteran, and a practitioner of an abstruse but essential craft called microcoding. He was soft-spoken. He had a mischievous air and—in all matters, it seemed—an aversion to the blunt approach. Alsing had joined the Eagle project without any coaxing, and was the only one of West's three lieutenants to do so. West regarded Alsing as one of the few people around Building 14 in whom he could confide, and for his part, Alsing, who was something of a watcher—a moviegoer—was fascinated by West, especially at the onset of the project.

"We need more bodies, Alsing," West said that spring. "Shall we hire kids?"

A famous computer engineer had remarked that he liked to hire inexperienced engineers fresh from college, because they did not usually know what was supposed to be impossible. West had heard the remark. He liked the sound of it. He figured, too, that "kids" would be relatively inexpensive to hire. Moreover, this could be another way of disguising his true intentions: who would imagine that a bunch of recruits could build an important new computer? To Alsing, the idea was vintage West. It looked risky and compelling. Alsing became the Eclipse Group's chief recruiter.

West and Alsing agreed that they would have to hire the very best of that year's college graduates, even though, they told each other, they might be hiring their own replacements, their own "assassins." That was all very well, but the demand for young computer engineers far exceeded the supply. What enticement could the Eclipse Group offer that companies such as IBM could not? Clearly, it had to be the Eagle project itself. It was thought to be a fine thing in the fraternity of hardware engineers to be a builder of new computers—in the local idiom, it was the "sexy" job—and, Alsing knew, most big companies just didn't offer recruits the opportunity to be such a person right away. So they had what West called "a high-energy story."

But the new recruits were going to be asked to work at a feverish pace almost at once, and they'd have no time to learn the true meaning of the Eclipse Group's mysterious rite of initiation, which was known as "signing up." In the Eclipse Group, when you signed up, you agreed to do whatever was necessary for success and to forsake time with family, hobbies, and friends—if you had any of those left, and you might not, if you had signed up for too many projects before. In effect, a person who signed up declared, "I want to do this job and I'll give it my heart and soul." Formal declarations weren't called for. A simple "Yeah, I'll do that" could constitute signing up. But only veterans knew what such a statement might entail.

The Eclipse Group solicited applications. One candidate listed "family life" as his main avocation. Alsing and another of West's lieutenants were skeptical when they saw that entry. Not that they wanted to exclude family men, being such men themselves. But Alsing thought: "He seems to be saying he doesn't want to sign up." The other lieutenant pondered the application. "I don't think he'd be happy here," he said to himself.

Any likely-looking candidate was invited to Building 14, and the elders of the group would interview the young man; it was usually a young man, for female engineers specializing in the hardware of computers were still quite scarce. If the recruit was a potential microcoder, his interview with Alsing was crucial. And a successful interview with Alsing constituted signing up.

Alsing would ask the young engineer, "What do you want to do?"

If the recruit seemed to say, "Well, I'm just out of grad school and I'm not really sure," then Alsing would usually find a polite way to abbreviate the conversation. But if the recruit said, for instance, "I'm really interested in computer design," then Alsing would press on. The ideal interview would proceed in this fashion:

"What interests you about computer design?"

"I want to build one," says the recruit.

"What makes you think you can build a new computer?"

"Hey," says the recruit, "no offense, but I've used some of the machines you guys have built. I think I can do a better job."

"Well, we're building this machine that's way out in front in technology," says Alsing. "We're gonna design all new hardware and tools. Do you like the sound of that?"

"Oh, yeah," says the recruit.

"It's gonna be tough," says Alsing. "If we hired you, you'd be working with a bunch of cynics and egotists and it'd be hard to keep up with them."

"That doesn't scare me," says the recruit.

"There's a lot of fast people in this group," Alsing goes on. "It's gonna be a real hard job with a lot of long hours. And I mean *long* hours."

"No," says the recruit. "That's what I want to do, get in on the ground floor of a new architecture. I want to do a big machine. I want to be where the action is."

"Well," says Alsing, pulling a long face. "We can only let in the best of this year's graduates. We've already let in some awfully fast people. We'll have to let you know."

"We tell him that we only let in the best—then we let him in," Alsing said, after it was all done. "I don't know. It was kind of like recruiting for a suicide mission. You're gonna die, but you're gonna die in glory."

GETTING A MACHINE OUT THE DOOR MAY HAVE BEEN the object of the team's labors. But the main source of motivation lay elsewhere. One day an old hand (old in a relative sense, since most in the team were very young) reflected on his job: "I said, 'I will do this. I want to do it. I recognize from the beginning it's gonna be a tough job. I'll have to work hard, and if we do a good job . . . we get to do it again.'" West called this "pinball." "You win one game, you get to play another."

Salaries within the Eclipse Group were respectable. At the start of the project, in 1978, a "kid" earned about \$20,000 a year. But engineers are regarded, and regard themselves, as professionals; therefore, they'd get no extra pay for overtime, no matter how many hours they worked. For previous jobs, old hands had received some stock options, and there was talk that the team's mem-

bers might get options if Eagle was a success. No one in authority actually stated that promise, it seemed. "But it sure as hell was suggested!" said one of the recruits. No one in the team said he expected to receive large amounts of stock, however. Most insisted that they weren't working on Eagle for money. When they talked about rewards, they spoke mainly of pinball.

By the fall of 1978, the preliminaries were complete. The kids—about a dozen of them—were hired, the general sign-up had been performed, and at least one possible reward, the Sisyphean one, had been clearly established. They had already begun to design the computer.

Eagle took its first material form in paper, in bound books as large as atlases, which contained the intricate geometric depictions of the circuits (the "schematics"), and in a fat volume of pages filled up with line after line of 0's and 1's—the microcode, the synaptic language that would fuse the physical machine with the programs that would tell it what to do. One could think of this small library of microcode and schematics as the engineers' collected but not wholly refined thoughts on a variety of subjects. The language was esoteric, but many of the subjects were as familiar as multiplication.

Chips, the product of the era of microelectronics, took most of the pure physics and plain electrician's work from the engineers' endeavor. Some circuit designers likened the chips to a collection of children's building blocks, which they had to assemble. Some referred to the entire realm of chip design and manufacture as "technology," as if to say that putting those chips together to make a computer was something else.

Many of the chips that Eagle's circuit designers used came ready-made to perform certain operations, such as addition. Others weren't completely ready-made, but in all cases one of the routine parts of the engineers' job was transferring their ideas to silicon and wire. The hardest part was concocting those ideas. Creating Eagle's hardware was primarily a matter of constructing long skeins of logical thought. Some indication of the complexity of the job lies in the fact that Eagle would contain thousands of chips. The designers had to figure out what each of those chips should do and how to connect it to the others so that Eagle could perform all the operations in its "instruction set," and some of those operations were very tricky ones. That was only half of the job. The microcode had to be written, too. Each operation in Eagle's repertoire would be performed at the direction of a "microprogram"; each microprogram would consist of one or more (usually more) "microinstructions"; and each microinstruction would consist of seventy-five discrete electrical signals. Thus, in order to equip Eagle to perform just one of its roughly 400 basic operations, the engineers had to plan in complete detail the passage of hundreds, sometimes thousands, of signals through the circuitry. They had to ensure, of course, that there was an absolute marriage between those signals and the circuits.

And they had to be sure that the performance of one operation did not foul up the performance of another. Possibilities for creating internal contradictions were numerous. The engineers had to try to anticipate all of them, and the difficulty of doing so was exacerbated by the fact that more than twenty people were creating this design in a hurry.

A brief description of the physical engine they were making could have been set to the music of "Dry Bones." This device was connected to that device, and so on. There was the microsequencer, which managed the microcode, sending it out to other parts of the engine on command from the instruction processor, or IP, which made assumptions about what basic chores the machine would be asked to perform in the future. There was the input/output controller, the IOC, which mediated between user and machine. The system cache was full of fast memory circuits and kept tabs on the IP. The address translation unit, ATU, kept track of the machine's main storage, and the control console (C/C), among other roles, acted as Eagle's therapist, by monitoring certain parts of the engine for problems and flaws, which are also known as "crocks." Practically every device relied upon the powers of the arithmetic and logic unit (the ALU), or "number cruncher," the heart of any computer; it did Eagle's math. There was also a clock, which ticked every 220 billionths of a second, telling all the rest of the machine that one microinstruction had ended and another had begun.

A great deal of the designing took place in silence, while engineers sat in their cubicles and paced in hallways and stood in their showers at home. It was the sort of work that was hard to escape, some said. One starts to imagine that trees and roads embody block diagrams and microprograms; this was the banal sensation that a member of the team had in mind when he said that it took three days for him to get Eagle out of his mind. During this time, West spent many hours in his office, staring at the team's designs of the circuitry. Usually, he drove away from Building 14 at high speed. "I can't talk about the machine," he said one evening, as he bent forward over the wheel. "I've gotta keep life and computers separate, or else I'm gonna go mad."

**S**OME OF THE YOUNG ENGINEERS WERE ASSIGNED TO work on microcode; they were called, and called themselves, "the Microkids." Those who went to work on the hardware, the actual circuitry, were known as "the Hardy Boys." This was the first real job for most of them. For some, at least, it was a strange beginning. Eager to make a good impression, and thinking it was the proper thing to do, one of the Hardy Boys set out, when he arrived, to meet his new team's leader. He went into West's office, extended his hand, and said, "Hi, I'm Dave." He would never forget that experience. "West just

sat there and stared at me. After a few seconds I decided I'd better get out of there."

For a Microkid named Jon Blau, the first months on the job seemed relatively serene. After Blau was assigned his cubicle, Alsing and the group's chief secretary helped him to find his way around the in-house computing system. Then Chuck Holland, a submanager under Alsing, drew for him a clear picture of the entire microcoding job. Holland had divided the overall task into several smaller ones, and he let Blau choose one from among them. Blau decided that he would like to write the code that would tell Eagle how to perform a lot of its arithmetic. He had always liked math, and he felt that this job would help him understand it in new, insightful ways. He spent most of his first months reading, in order to prepare himself. All in all, life in the Eclipse Group did not seem very different from life at his alma mater, MIT.

Then one day in the fall of 1978, Blau was sitting in his cubicle studying Booth's Algorithm, a procedure for doing multiplication, and he was thinking to himself, "This is pretty slick," when Alsing poked his head around the partition and said, "There's a meeting."

Blau trooped into a conference room with most of the other new members of the team, joking and feeling a little nervous. There waiting for them were the brass: the vice president of engineering, another lower-level but probably important executive, and West, sitting in a corner, chewing on a toothpick. The speeches were brief. Listening intently, Blau heard all about the history of 32-bit minicomputers. They were really catching on, and the word was that DEC would probably introduce a new model of VAX in about nine months. Eagle was already late. It had to be designed and made ready for market by April, in just six months. That wouldn't be easy, but if any engineers could do it this group could, the bosses said.

Blau felt proud of himself and pleased about this first real job of his when he left that meeting. He went right back to his cubicle, and picked up Booth's Algorithm. Then, suddenly, he felt it, like a little trickle of sweat down his back. "I've gotta hurry," he said to himself. "I've gotta get this code written today. This is just one little detail. There's a hundred of these . . ."

Practically the next time he looked around him it was midnight, but he had done what he'd set out to do. He left the basement thinking, "This is life. Accomplishment. Challenges. I'm in control of a crucial part of this big machine." He looked back from his car at the blank, brick, monolithic back of Building 14 and said to himself, "What a great place to work." Tomorrow he would have to start encoding an instruction called FFAS. He told himself that this wouldn't be too hard. When he woke up the next morning, however, FFAS was upon him. "Oh, my God! FFAS. They need that code next week. I'd better hurry."

"The pressure," Blau later recalled. "I felt it from inside of me."



**A**ROUND THIS TIME, A HARDY BOY NAMED DAVE EPSTEIN was dreaming up the circuits of the device called the microsequencer. No other parts of Eagle could begin to function without it, so the manager of the Hardy Boys, Ed Rasala, wanted it designed quickly. "How long will it take you?" Rasala asked Epstein.

Epstein replied, "About two months."

"Two months?" Rasala said. "Oh, come on."

"Okay," said Epstein. "Six weeks."

"I just wrote my own death warrant," Epstein thought. Six weeks didn't look like enough time. He took to staying at his desk half the night, and the microsequencer took shape more quickly than he had expected. He felt so happy that he went down the hall and told Rasala, "Hey, Ed, I think I'm gonna do it in four weeks."

"Oh, good," Rasala said.

Epstein returned to his cubicle. Then he realized what had happened. "I just signed up to do it in four weeks."

Afterward, Epstein remarked, "I don't know if I'm complaining, though. I don't think I am. I work well under pressure."

Not everyone was stimulated. One newcomer was astonished at the way the team was being managed. Hardy Boys and Microkids were making deals, saying to each other, in effect, "I'll do this function in microcode if you'll do this one in hardware." He was a little older than the other newcomers and had some experience in computer design, and he had never seen it done this way. "There's no grand design," he said. "People are just reaching out in the dark, touching hands." He was having some problems with his own part of the design and he felt sure that he could solve them properly if the managers would simply give him time. But they kept saying there was no time. No one seemed to be in control. Nothing was ever explained. The team's leader rarely even said hello to his troops. Make a mistake, however, and the managers came at you from all sides.

"The whole management structure . . ." said this young engineer. "Anyone in Harvard Business School would have barfed."

If West had heard that remark, he might have taken it as a compliment. Carl Alsing had often heard West use the phrase "flying upside down." The inspiration for it

evidently came from a friend of West's who used to do that very thing in his airplane. By the term, West seemed to mean the assumption of large risks, and the ways in which he applied it left Alsing in no doubt that flying upside down was supposed to be a desirable activity, the very stuff of a vigorous life.

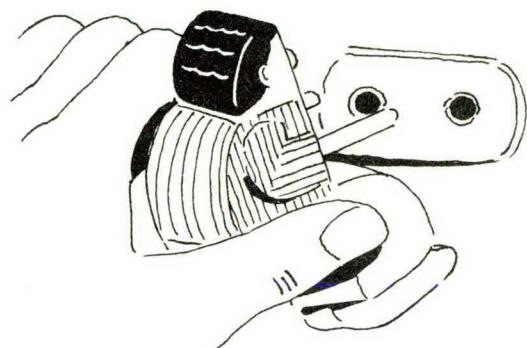
Ed Rasala acknowledged that West made a project more dramatic, "definitely more dramatic" than it had to be. Rasala smiled at the thought, however, and he did believe it when West said that they had to fly upside down in order to ship Eagle on time. As for Alsing, he admired West's style: "I screamed and hollered over NAND gates and microinstructions when we built the first Eclipse, but I'm too old to feel that way about computers now. This would be crashingly dull if I was doing it for someone else. West is interesting. He's the main reason why I do what I do."

First coaxing, then threatening, saying, "If you don't do this, your job description is inoperative," West had persuaded a senior engineer named Steve Wallach to work on Eagle. Wallach was a combative engineer, in his thirties. West called him "a walking dictionary and encyclopedia of computers." Wallach had not liked the looks of this machine; he had called it "a wart on a wart on a wart" and "a bag on the side of the Eclipse." But as he got deeply involved, he began to like the looks of Eagle. He was the machine's chief architect. One can think of a computer's architecture as a description of what the machine will do; it leaves out the intricate details of the computer's construction. But Wallach, who was doing a fine job, in West's opinion, wasn't really finished before West had the team designing printed circuit boards. Before the team could clean up its designs, West was ordering prototype boards. Before the prototype boards could possibly be perfected, West was arranging for the construction of final, etched boards. And long before they could know whether Eagle would ever become a functioning computer, West had the designers stand in front of a camera and describe the various parts of the machine. The result of this last act of hubris (there were many others) was a videotape some twenty hours long. West planned to use it, when the right time came, as a tool for spreading the news of Eagle all around Building 14. "Pretty gutsy," he said, gesturing at the canisters of videotape in his bookcase.

West maintained that the team had to show quick and constant progress, whether or not any progress was being made, in order to get the various arms of the company interested in helping out—in order, that is, to compete successfully for resources. One evening he offered an additional explanation. "I'm flat out by definition. I'm a mess. It's terrible," he said. He paused, and added, "It's a lot of fun."

Others echoed that remark. The team was working long hours now, often twelve a day, and usually they put in six days a week. They spoke about the harsh effects of

such labor, but there was relish in their voices. One day Jon Blau said, "I've had difficulty forming sentences lately. . . . Pieces of your life get dribbled away. I'm growing up, having all those experiences, and I don't want to shut them out for the sake of Data General or this big project." But he said that on the whole he was happy. He added, "That's the big kick, that the guys with the purse strings are trusting a bunch of kids to come up with the answer to VAX. That's what bowls me over, that they haven't just put us in a corner somewhere, doing nothing."



ALTOGETHER, IT TOOK THEM ABOUT SIX MONTHS TO dream up Eagle. By January of 1979, two partially assembled prototypes were sitting in the basement, behind the locked doors of a small lab. But Eagle was not yet the equal of a hand-held calculator. The team had to make the computer work. They had to test it, and find, identify, and repair the flaws in its design. They called this part of the project "debugging."

Eagle had been designed too quickly for prudence, and at the onset of the debugging the question arose, in West's mind especially, whether "fatal" flaws in the design were about to be revealed. Pursuing what he called "what's-the-earliest-date-by-which-you-can't-prove-you-won't-be-finished scheduling," West had promised his bosses that Eagle would be debugged and brought to life by April. Of course, one can't ever know how long a debugging will take; the debugger delves into uncertainty. West was very nervous. He had Rasala make up a debugging schedule that would bring the machine in by April. Rasala put the Hardy Boys on two shifts in the lab. They made some progress right away, and West felt relieved. Then progress all but ceased. In the local idioms, they were moving "three steps forward, two steps back" and the debugging schedule was "slipping a week a week."

Right around the time when West was brooding over these discouraging reports from the lab, he got word from his boss upstairs that the company team in North Carolina was going to miss its deadline by a considerable margin. This was momentous news. West had always maintained that Eagle was crucial to the company. Now he was being proved right. He did not feel like celebrat-

ing, though. At the moment, he was afraid that Eagle might never get debugged, and yet he was being told that it had to be done within four months. Everyone in the company was depending on the Eclipse Group now; that was the message West thought he received from upstairs. Carl Alsing thought it might be the message that West decided to receive. "If you say you're gonna do it in a year and you don't take it seriously, then it'll take you three years," Alsing explained. "The game of crazy scheduling is in the category of games that you play on yourself, in order to get yourself to move."

**A**N ENGLISHMAN OF THE NINETEENTH CENTURY named Charles Babbage, who is best known as the father of the modern computer, was one of the first to express the principle that the way to get a job done cheaply is to divide it into small parts that require relatively unskilled and therefore inexpensive labor. This is a basic principle of modern industrial organization. The general problem that it causes, one that has kept many consultants and psychologists busy over the years, is that when jobs are reduced to tasks so small and simple that the people performing them scarcely have to think, then work becomes boring and workers alienated. Most jobs were boring, in the computer industry and elsewhere, Alsing suspected, and he'd heard of systems of management that made even the building of a new computer sound dull. West, by contrast, had an exalted notion of what it meant to build a new machine; he spoke of "getting way out on the edge of what your mind can comprehend." He said, "It's for the kind of guy who likes to climb up mountains." According to the theory that West expressed to Alsing, computer engineers—the "winners" among them, anyway—would motivate themselves and work passionately, their only reward pinball, if they were given real responsibility for creating a new machine. That was how Eagle was being managed.

In West's lexicon, these ideas were subsumed in the word "trust." He'd added the term to his vocabulary that fall, and the ways in which he used the word made Alsing wonder if either of them had ever heard it before. "Trust is risk, and risk-avoidance is the name of the game in business," West said in praise of trust. Although he scarcely seemed to notice them, West often bragged about the engineers on his team behind their backs, and he could draw convincing portraits of most. He believed that they were bound together by webs of mutual *trust*. They'd signed up to do a job for him, and he in turn *trusted* them to get it done. He did not attempt to keep complete control, by breaking every task into meaningless fragments.

But the debugging wasn't going well, and in the circumstances, watching from his lair of an office was strenuous work for West. Alsing had known West back in the days when West was an engineer and not a manager.

In Alsing's opinion, West had been just a competent circuit designer, but he was a very adept debugger. West wanted to be a debugger again. He wanted to go into the lab and will that machine into life, Alsing thought. But if West barged in there now, he would be admitting that he didn't really trust his team. The magic would be lost. So West was staying away from the lab and worrying instead.

Almost every morning now, West called Alsing into his office, closed the door, and asked, "What's really going on in the lab, Alsing?"

Alsing made soothing replies. Consoling the boss, it seemed to him, had become part of his job, and not necessarily the easiest part.

The tension among the recruits seemed palpable. Alsing could feel it in the air, he said. One day, early in the debugging, in a cubicle right outside West's door, a couple of Microkids began to laugh—first one, then the other, then both of them, more and more raucously. Alsing was sitting nearby, in his own cubicle. He heard the laughter. In the midst of it, his phone rang.

It was West. "If you don't shut those guys up I'm gonna kill 'em!"

Alsing went out and asked the offending Microkids to do their laughing elsewhere. "It was awful," he said afterward. "I felt so embarrassed. I felt like one of those old supervisors from the 1800s who used to hire children and work them eighteen hours a day."

West took a day off and drove to the seacoast, to look at sailboats.

In the evenings, West would usually call a halt to flying upside down and climb out of the role of the tough, mean manager. He would leave his office door ajar, as an invitation, and, leaning back, his hands fallen still, he would entertain almost any visitor. But on the evening when West heard the news about North Carolina, nightfall seemed to bring him no relief. West had a wild air, as if his office were a cage. He talked on and on. He had grown perceptibly thinner in the past few months, and his hands looked outsized. As he spoke, he pushed his hair back, he drove his index fingers up under the bridge of his glasses, he made fists and exploded his fingers outward. His hands, one imagined, had primitive will; they wanted to get into the prototype Eagles.

West said that he had been told the company would be "in a lot of trouble" if Eagle weren't debugged by April. "Suppose I quit?" he asked. "I could just say, 'The hell with it,' and go . . ." Maybe he would buy a boat and sail away from Data General. No, he wouldn't do that yet. He'd see this project through, but only this project. "I'm not gonna do the next machine. I'm gonna give somebody else the chance to fail. I'm gonna get totally out of computers."

Saying he would quit was for West the equivalent of saying that he sometimes felt like quitting, and the statement was a substitute for the act itself.

THE LAB WAS BORROWED SPACE, A CORNER OF A LARGER lab, sealed off by a thin steel partition. It was no bigger than most suburban living rooms and more crowded than most. Along one cinder-block wall stood the source of anxiety, a pair of Eagles. They were two bare metal frames, their tops about shoulder-high. Inside each frame, exposed to view (computers in this state are said to have their "skins" off), was a shelf full of boards. Small flat cables ran among the boards, and further inside, below the shelves, were many bundles of multicolored wire. Eagle was a tangle of wire. A system console, which looked like a large typewriter, attended each prototype, and a magnetic tape drive stood next to one. The tape drive is often shown on TV and in movies, in order to signify the presence of a working computer. Probably it is chosen because the reels of tape spin rapidly and thus prove that something is going on. In fact, though, tape drives are among the slowest parts of a computing system. The real action takes place inside and between the boards. To get a look at it, one needs special tools—boxy little machines, covered with switches, called "logic analyzers." They sat on low, wheeled carts. Each analyzer had a small screen. In essence, one engineer explained, the analyzers were cameras. They took pictures of what happened inside the computer. Eagle would do a cycle of work every 220 nanoseconds, or billionths of a second. Hooked up to some part of the machine, an analyzer could take pictures of what happened there in each of 256 cycles and play those pictures back on command. "It's funny," said Ed Rasala. "I feel very comfortable talking in nanoseconds. I sit at one of these analyzers and nanoseconds are *wide*. I mean, you can see them go by. 'Jesus,' I say, 'that signal takes twelve nanoseconds to get from there to there.' Those are real big things to me when I'm building a computer. Yet when I think about it, how much longer it takes to snap your fingers, I've lost track of what a nanosecond really means." He paused. "Time in a computer is an interesting concept."

One evening, when the lab was very still, everyone there bent to some task, one young engineer fiddled with a prototype Eagle. He was having a problem with the board he had designed. The machine, he said, kept "going to never-never land" whenever he asked it to add, and he was trying to take a picture with an analyzer of what was going wrong inside.

A straight white line ran horizontally across the analyzer's small blue screen. Nibbling his nails absentmindedly, the engineer made some arrangements around the

prototype Eagle. Then, his nails still at his lips, he turned to the analyzer.

Something had happened. The straight white line on the screen had rearranged itself into a jagged shape. The young engineer stared at this picture. Slowly, he rotated his hand and took most of his knuckles in his teeth.

The jagged line on the screen was a picture of an electronic event that had taken place, in infinitesimal time, just a moment before. Though it was a common sort of picture in the lab, all of a sudden it looked dreadful.

This young engineer was, as it happened, one of those who did not enjoy flying upside down. He was capable, a Microkid said, of coming up with remarkable insights about the innards of a computer; there was no doubting his abilities. But he'd had trouble with the team's top managers and with his piece of Eagle's hardware. Many of the recruits felt that they were being given substantial freedoms. Clearly, though, time to revise was not one of them. This engineer was unhappy because he had been denied the right to fix the problem in his board in the way he thought was proper. In addition, he felt he had been insulted on a number of occasions. Perhaps most important, he felt worn out.

As the debugging continued, he felt under extreme pressure, which collected in his stomach. It hurt every day. This sort of work, even the occasional bad stomach, used to be fun. "Part of the fascination," he said, "is just little boys who never grew up, playing with erector sets. Engineers just don't lose that, and if you do lose it, you can't be an engineer anymore." He went on, "When you burn out, you lose enthusiasm. I always loved computers. All of a sudden I didn't care. It was all of a sudden a job."

One weekend some time before, he had visited what he called "a *very* liberal arts college" in Vermont. He was strolling across the campus when a young woman, bare to the waist, walked by. "She was a miracle of biological engineering," he said. "I was so stunned that I walked into the door of a geodesic dome. Although blood was pouring down the bridge of my nose, I was completely oblivious to it."

Back at Data General, one day during the debugging, his weariness focused on the logic analyzers and the small catastrophes that come from trying to build a machine that operates in billionths of a second. He went away from the basement of Building 14 that day, and left this note in his cubicle, on top of his computer terminal: "I'm going to a commune in Vermont and will deal with no unit of time shorter than a season." □

Copyright of Atlantic Magazine Archive is the property of Atlantic Monthly Group LLC and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.