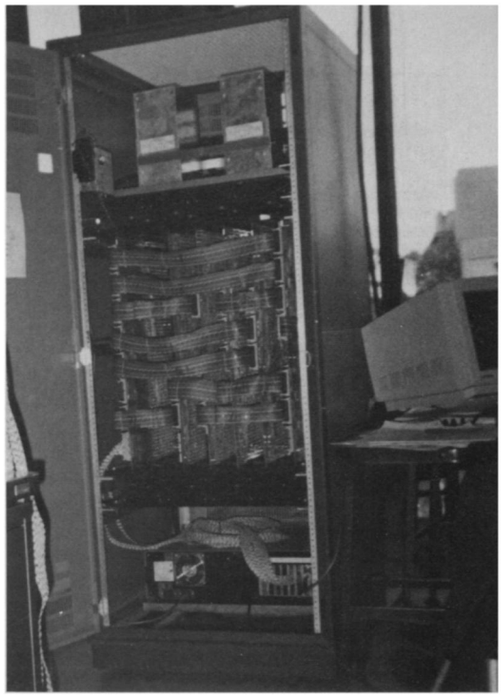
Little Fermat

A scheme for speeding up multiplication leads to a unique computer

By IVARS PETERSON



Multiplying million-digit numbers takes time — lots of time. Even today's supercomputers are poorly equipped for efficient, errorless number crunching on such a scale. Nonetheless, many mathematical and scientific applications, from identifying prime numbers to modeling weather patterns, require large-number computations.

Is there a faster way of multiplying gigantic numbers? Nearly four years ago, M.M. (Monty) Denneau of the IBM Thomas J. Watson Research Center in Yorktown Heights, N.Y., and mathematicians David V. and Gregory V. Chudnovsky of Columbia University in New York City decided there really is, and they designed a new machine to prove their point.

The resulting computer, painstakingly assembled from commercially available parts by MIT graduate student Saed G. Younis, now stands nearly 6 feet tall in a laboratory and ready to take on the world. Dubbed "Little Fermat," after the 17th-century French mathematician Pierre de Fermat, it works with instructions and data expressed in 257-bit "words" and uses a special kind of arithmetic based on so-called Fermat numbers. These characteristics clearly differentiate the new machine from conventional computers.

"Little Fermat is a high-performance, general-purpose scientific computer," David Chudnovsky says. Its novel features make it particularly efficient for solving a variety of numerical problems ordinarily plagued with errors because of the way conventional computers express and round off numbers.

"There's no machine like it in the world," Gregory Chudnovsky asserts. Indeed, he adds, Little Fermat vividly demonstrates the kinds of capabilities that could enhance the performance of future supercomputers.

Using pencil and paper, human beings can add, subtract, multiply or divide numbers of any length, albeit slowly. Computers, on the other hand, are designed to manipulate numbers of a fixed length. For instance, simple personal computers typically work with digit strings, or words, that consist of eight digits, or bits, each bit being a one or a zero. Today's most advanced supercomputers handle 64-bit words.

By using longer words, a computer can calculate with greater precision and make finer distinctions when converting, say, an audio signal into strings of digits. For example, an 8-bit signal processor divides an audio signal into at most 256 intensity levels, providing a relatively crude approximation of the original waveform. In contrast, a 16-bit signal processor — the sort used to record music on compact disks — samples many times more levels, producing digital audio signals of significantly higher quality and much less distortion.

In scientific computations, the loss of precision caused by using shorter words can have serious consequences. Many physical processes, such as the flow of water past a ship's hull, are full of inherent instabilities. When a computer simulates such processes, it must perform trillions of arithmetic operations. Even a slight inaccuracy in the description of how a physical system changes over time, or in rounding off numbers during a computation, can lead to the wrong answer.

But the penalty for increased word length is a corresponding increase in the amount of circuitry and wires needed to build the computer and in the time the computer takes to execute an instruction. Schemes that allow small-word computers to handle longer words circumvent the problem, but such hybrid operations generally prove astonishingly slow and cumbersome.

To fit more numbers into a given word length, computer scientists over the years have developed special formats for representing decimal or real numbers in a computer, along with specific rules for rounding off or truncating such numbers to make sure they stay within the assigned word length. Most computers now use such "floating-point arithmetic" schemes for representing and manipulating numbers. But small errors inherent in the way real numbers are represented in a computer can accumulate, sometimes causing major precision problems in numerical calculations.

Number theory offers a way to rid calculations of these intrinsic errors by combining a special procedure called modular arithmetic with a set of numbers known as Fermat numbers.

In modular arithmetic, only remainders left over after division of one whole number by another count. For example, suppose the divisor, or modulus, happens to be 5. Dividing 5 into a given whole number produces a certain remainder, which constitutes the answer. Thus, dividing 5 into 7 or 12 produces the same answer — the remainder 2.

Fermat numbers have the form $2^x + 1$, where $x = 2^n$. When $n = 0$, the first Fermat number, F_0, is 3; when $n = 1$, the second Fermat number, F_1, is 5; similarly, $F_2 = 17$; and so on (SN: 6/23/90, p.389). Using a Fermat number as the divisor in modular arithmetic provides a handy way of speeding up certain types of calculations and circumvents the need to deal with real numbers.

In 1975, James H. McClellan of MIT's Lincoln Laboratory in Lexington, Mass., built a digital signal-processing device based on Fermat arithmetic, demonstrating that the electronic circuitry needed to do modular arithmetic based on Fermat numbers can operate faster than the circuitry used for performing real-number operations. Furthermore, no rounding off takes place during the calculations. Thus, the answer is always exact and correct, provided it's less than the Fermat number used in the operations.

Little Fermat's answer to achieving faster multiplication while avoiding the errors associated with floating-point arithmetic is to combine increased word length with numerical recipes, or algorithms, based on modular arithmetic and Fermat numbers.

Armed with these key ideas, Denneau and the Chudnovsky brothers prepared a flow chart, then a detailed design for a machine

capable of rapid, error-free multiplication of large numbers. Then the real headaches began.

Commercially available integrated-circuit components limited the word size to 257 bits. Wiring constraints restricted the size of the boards on which the electronic parts could be mounted. Instead of laying out the computer on a single circuit board, the designers had to break up the circuitry to fit onto six boards – each a square 25.6 inches wide, densely packed with chips and covered with a rat's nest of connecting wires.

Before Younis could set the first chip into place, the researchers had to check their design for flaws. The trouble was that they had designed Little Fermat to have capabilities exceeding those of any conventional computer that could be used to simulate the way its logic worked. In the end, they had to settle for testing their design in pieces, never as a complete unit.

"Even then, it was a staggering task," Gregory Chudnovsky says.

Younis spent more than a year building the computer, then roughly another year testing the completed machine to correct all the assembly and design defects that he found. The biggest assembly problems involved the 82,500 individual wires (totaling about 5 miles) connecting 6,700 integrated-circuit chips and other components.

Those problems ranged from chips that sporadically continued working even when no electrical power reached them to wires that shrank and disconnected when they cooled after the machine was turned off. And because the computer was designed for rapid calculation, and electronic signals travel at finite speeds, even wire length became an important consideration. The most nightmarish defects – especially those that made their presence felt intermittently – took weeks to track down, but Younis persisted.

"Now it's running," David Chudnovsky says. "Rarely has a hardware project of such magnitude been carried through to its completion by a single man. It was an unbelievable achievement."

To compute with Little Fermat, a user writes a program in a language now called Younis. That language provides a set of instructions expressed in 240-bit chunks, which can be combined in various ways to perform a number of functions. A personal computer attached to Little Fermat loads the program into the machine, monitors the computation and unloads and displays the results when the computation is finished.

"We are now checking [Little Fermat's] performance," Gregory Chudnovsky says. "We have to be sure it does what we want it to do. And we would be happy to find someone interested in programming the machine for a specific application."

So far, the Chudnovskys have used Little Fermat primarily for computations in number theory that involve gargantuan numbers – searching for prime Fermat numbers, factoring large numbers and testing whether certain huge numbers are primes.

But the machine's special characteristics make it ideal for digital signal and image processing, as well as for solving the differential equations used by researchers modeling the behavior of physical systems. Such computational problems regularly surface in aerodynamics, hydrodynamics, chemistry, geophysics and many other disciplines.

Only one Little Fermat exists today, but that's more than can be said for the many other new computer designs that never made it to the hardware stage, instead remaining "paperware" – described in a paper but never built. "This machine is alive and well and working," David Chudnovsky says. "It's real."

"We showed it can be done," Gregory Chudnovsky says. "Even if it remains a one-of-a-kind machine, Little Fermat stands as a demonstration of what should be added to a supercomputer to improve its performance. It would be very cheap to put additional Fermat circuitry into future supercomputers." □

Continued from p. 219

Across the country, in the deserts of New Mexico, Thomas E. Hakonson is testing landfills designed to stay bone dry. As manager of environmental sciences research at Los Alamos (N.M.) National Laboratory, Hakonson strives to eliminate any chance of leachate formation in the experimental dump sites he's designing. His approach is to capture any precipitation at the surface, before it penetrates the landfill, and to return that water to the air via evapotranspiration by plants.

Using computer models, Hakonson also analyzes data on water migration, soil composition and erosion to design sloping gravel covers that wick moisture laterally away from wastes. Large rocks buried below topsoil prevent burrowing animals and roots from penetrating wastes buried below. But a key to these garbage vaults is the covering of native foliage planted atop them. Hakonson selects the plants for their ability to drink up rain or dew.

Systems that rely on synthetic flexible liners to entomb wastes eventually break down, Hakonson observes. His design "is less prone to failure because it uses natural components," he says. "We've got forests of juniper trees, grasses and weeds." Such environments are very stable, he adds: "They've been here hundreds of years."

Though officials at the Los Alamos facility would like to use these structures for hazardous waste disposal, Hakonson says his landfills could just as readily store municipal garbage.

Preliminary studies indicate the newly designed waste sites offer safe, long-term storage in the dry Southwest. To learn whether they will provide comparable security against leakage in wetter, colder climes, Hakonson has set up experimental models at Hill Air Force Base in Ogden, Utah. This should prove a true challenge of the system's universality, he says: "Snow comes in the winter and melts in the spring [when plants aren't transpiring]," he says, "so the mechanisms for removing water are low."

Even conventional dry landfills can benefit from better techniques and materials, says civil engineer Robert E. Landreth, chief of landfill technology at EPA's Risk Reduction Engineering Laboratory in Cincinnati. For example, instead of burying each day's accumulation of wastes under several inches of soil, landfill managers can preserve space by blanketing wastes overnight with synthetic covers, such as a layer of foam. The next day, bulldozers break the foam moisture barrier before the next load of waste arrives.

Unlike conventional landfills, which permanently segregate daily garbage deposits in dirt-shielded cells, these allow the interred wastes to mix into a more homogeneous mass and accelerate decomposition, says Landreth.

To determine decomposition rates in full-scale wet and dry landfills, University of Wisconsin civil engineer Robert Ham plans to analyze working landfills in Florida, New York, Pennsylvania and Wisconsin. The Wisconsin landfill will include experimental wet and dry cells; the others will have sections covered with sand, allowing precipitation to flow into the garbage. Results of his work won't be known until the mid-1990s.

Though wet landfills must be monitored more carefully than dry ones, Ham says they can be cost-effective in the long run by shortening the decay period and thus reducing the time required for monitoring. Before landfill managers recognized the extent of the leakage problem, "you'd finish up the landfill, cover it and walk away from it," he says. But those days are over. "What we're talking about is getting the bulk of decomposition to occur more rapidly so we don't have exposure to problems many years in the future." □