MATHEMATICS

How Number Theory Got the Best of the Pentium Chip

Chalk one up for number theory. With lurid accounts of the flaw in Intel's Pentium processor making front-page and network news, users of the personal computer chip in fields ranging from science to banking are finding cases where its faulty logic sends their computations awry. But the problem might have gone undetected for much longer if the chip had not slipped up months ago during a long series of calculations in number theory, raising the suspicions of a dogged mathematics professor.

To other mathematicians, the discovery of the flaw by Thomas Nicely of Lynchburg College in Virginia emphasizes the value of number theory—the study of subtle properties of ordinary counting numbers—for providing quality control for new computer systems. By forcing a computer to perform simple operations repeatedly on many different numbers, number-theory calculations "push machines to their limits," says Peter Borwein of Simon Fraser University in Burnaby, British Columbia. Many computer makers have adopted these calculations as a shakedown test for systems intended for heavy-duty scientific computation, and although the practice has yet to spread to personal computers, Borwein and some other mathematicians think that might be a good idea.

Intel had actually found the flaw by other means after the chip had gone into production, but had decided that it was not likely to affect ordinary users. But the company hadn't counted on the use that Nicely had in mind. When he fired up a Pentium computer last March, Nicely was adding its number-crunching power to a project in computational number theory he had begun the year before. He was trying to improve on previous estimates of a number called Brun's sum, which is related to the distribution of prime numbers.

The sequence of prime numbers—2, 3, 5, 7, 11, 13, 17, 19, etc.—is a continuing source of fascination to mathematicians. Since the time of Euclid, they have known that there are infinitely many primes, but although primes are relatively abundant early on, they become scarce among larger numbers. For example, roughly 23% of two-digit numbers are prime (21 of 90), but the figure for ten-digit numbers is just 4%, and among hundred-digit numbers, the fraction of primes is less than half a percent. As a consequence, the gap between consecutive prime numbers tends to increase. However, every so often two odd numbers in a row turn out to be prime: 3 and 5, 41 and 43, 101 and 103, and 10,007 and 10,009, for example.

Mathematicians conjecture that such "twin primes" pop up infinitely often. But in 1919, the Norwegian mathematician Viggo Brun proved that even if there are infinitely many twin primes, the sum obtained by adding their reciprocals—the sum $(1/3 + 1/5) + (1/5 + 1/7) + (1/11 + 1/13) + \ldots$ —converges to a finite value, much as the sum $1/2 + 1/4 + 1/8 + 1/16 + \ldots$. converges to 1. Brun's sum is known only to the first few digits, however—and even there, the accuracy is based on conjectures about the frequency with which twin primes occur. Number theorists think it's unlikely that clumps of twin primes are lurking among very large numbers, but they have been unable to prove it. One way to check up on this assumption is to compute better estimates for Brun's sum.

"In desperation, I ran this portion of the calculation on one of the 486s. ... The error disappeared."

—Thomas Nicely

LYNCHBURG COLLEGE

In 1974, two mathematicians working for the Navy, Daniel Shanks and John Wrench Jr., reported the first computationally intensive estimate of Brun's sum, based on the occurrence of twin primes among the first two million prime numbers. Two years later, Richard Brent at the Australian National University calculated all twin primes up to a hundred billion (224,376,048 pairs), from which he computed an estimate of 1.90216054 for Brun's sum.

And there it sat—until Nicely entered the picture. The Lynchburg math professor decided to push Brent's work into the trillions. To be on the safe side, he computed Brun's sum twice, using two different methods: the "easy" way using a computer's built-in floating point unit, which is supposed to be accurate to 19 decimal places, and the "hard" way using an extended precision arithmetic, which he set to give 26 (and later 53) digits of accuracy. (The difference can be likened to the difference between computing $1/3 + 1/7$ as $0.33 + 0.14 = 0.47$ and computing it as $1/3 + 1/7 = 10/21 = 0.48$. The latter calculation gains accuracy by doing some exact arithmetic first.)

The comparison between the two methods is what got Intel into trouble. After Nicely added the new Pentium to his stable of computers, he found that the gap between the two results was much larger than it should have been. By trial and error and a process of elimination, he pinpointed the source of the problem: The Pentium was giving incorrect floating point reciprocals for the twin primes 824,633,702,441 and 824,633,702,443—they were wrong from the 10th digit on. Nicely still didn't know whether the error was caused by his hardware or software, in part because he'd caught an earlier error in a compiler program. "Finally, in desperation, I ran this portion of the calculation on one of the 486 [computers], rather than the Pentium," he recalls. "The error disappeared."

Even that didn't prove conclusively that it was the Pentium chip's fault; other hardware in the computer could have been responsible. But in October (4 months after he first noticed his calculations were off), Nicely nailed the culprit when he got hold of two other machines with Pentium chips and was able to reproduce the error. He notified Intel and, after getting no satisfactory answer by the end of the month, sent e-mail asking others to double-check his discovery. "I believe you are aware of events from that point on," he concludes dryly.

The Pentium's problem, as others have abundantly confirmed, lies in the way the chip does division. Although it works fine for most numbers, the chip's built-in algorithm makes mistakes in certain cases, rather like a grade-schooler who has mismemorized part of a multiplication table. Nicely estimates that the chip gets roughly one in a billion reciprocals wrong. But because the work in number theory required him to compute billions of reciprocals over a wide range, he was almost bound to run into the mistake.

"We've known for a long time that number theory computations are very helpful" for turning up computer errors, notes computational number theorist Arjen Lenstra of Bellcore, in Morristown, New Jersey. "It is useful to run number theory stuff on your processor before you sell it."

Intel hasn't decided whether to make such computations a routine part of its testing procedure, says Stephen Smith, engineering manager for the Pentium processor division. But Intel was so impressed with Nicely's work that it asked him to run further computations on a corrected chip. "We looked at him as the most thorough tester," says Smith.

–Barry Cipra