Pentium Division Bug

Sample Paper

October 30, 1994 a professor of mathematics at Lynchburg College named Dr. Thomas R. Nicely made a public announcement that he had discovered a flaw in the Pentium I chipset released by Intel. He noted that the floating point unit would return erroneous values for particular division operations. This incorrect result could be reproduced on any machine that contained the processor with the bug in any number of programs. This means that it was possible to test for the error in compiled code, any ordinary spreadsheet program such as Excel, or the basic operating system calculator. Nicely had discovered this flaw in June 1994 but did not feel confident in diagnosing the problem until he had conducted further testing. On October 24, 1994 he contacted Intel with the information about the problem. Nicely says in his public memo, "The contact person later reported that the bug was observed on a 66-MHz system at Intel, but had no further information or explanation, other than the fact that no such bug had been previously reported or observed."

The best example of when the chip would produce a very significant error can be found in the division of two large numbers. In the formula z=x-(x/y)*y where 'x' equals 4195835 and 'y' equals 3145727 the correct answer should be exactly zero. The affected Pentium chip would produce 256 as the correct answer. While this did happen, this was an extreme case and most erroneous results were only off by a few bits. Intel's initial claim was that after months of testing it was determined this error would only occur once in about nine billion division operations. Intel had tested the occurrences of the bug by using completely random numbers. In the first article actually covering the incident, Intel explains that this would not affect the common computer user at all in their work. An Intel spokesman said of Nicely "He's the most extreme user. He spends round-the-clock time calculating reciprocals. What he observed after running this for months is an instance where we have eight decimal points correct, and the ninth not showing up correctly. So

you get an error in the ninth decimal digit to the right of the mantissa. I think even if you're an engineer, you're not going to see this." Intel did make an immediate fix to all processors prior to their shipping after the bug was made public but they did not correct it when they discovered small errors in their labs prior to Nicely's memo.

Intel attributed the bug to a very subtle problem in a specific algorithm that closely resembles long division. The SRT algorithm is the one used for floating point division on the Pentium chip. Instead of using base ten decimals it uses base 4. This serves for very quick performance during large division computations. A triangular array with stored values is used to accomplish its calculations. The problem occurred when the SRT algorithm was implemented in silicon. Five values out of 1066 in the table were dropped, resulting in empty cells. These five were supposed to contain the value positive two but since there was no entry in those spaces any time an attempt to fetch information from them would be made zero would be returned. This caused an issue where the remainder may have been calculated slightly wrong but due to that value being carried forward it may increase the rate of error in the computation.

Intel's initial assessment using completely random numbers was accurate in producing only one error in nine billion division operations. However, their claim that it would only affect very specific users may not be entirely accurate. A study done by IBM produced results quite differently from those that Intel had released. Using their logic (which seems much more likely than Intel's) a problem occurs on the average once in one hundred million division operations. Not only was this error more likely but IBM also questioned Intel's assumption that the average user only did 1000 divides a day. They cited independent studies that showed a spreadsheet user completing common tasks doing 5000 divides a second. In only 15 minutes that is 4.2 million divides. These odds affect the average user much more than Intel initially speculated. It was also noted that numbers and calculations used commonly during financial calculations produced erroneous results more often than random numbers. This means many common users will encounter the problem several times a day. This would have a tremendously adverse affect on everyday life given the need of these programs by common users.

The accuracy of the calculations is no doubt a problem that required a resolution immediately. One of the major mistakes that Intel made was that they did not confront this problem when they had initially discovered it. They waited until a big deal was made about it and they completed a misleading study to avoid replacing many faulty processors. They initially decided to treat every claim on a case by case basis while denying common users a replacement. This

is a bad business practice to sell a faulty piece of equipment to a user and then refuse to fix it unless they only did complex mathematical operations on a daily basis. They did not change this policy until IBM decided to quit selling this particular chip. Intel claimed that IBM was overreacting with their study but conceded to replace any customers chip if asked.

This type of error may not seem like a big deal if finances are miscalculated but the magnitude of the problems it can cause are much greater than one might notice. On June 4, 1996 the Ariane 5 rocket exploded 37 seconds after liftoff due to an inaccuracy in a number. When a floating point was converted to an integer it was too large to be stored in the allocated space. The computer cleared its memory and instructions were given to its rocket nozzles, which caused an explosion.

Design in the computer world is very important but there will undoubtedly be mistakes. Intel's mistake was not so much in making the mistake since it was almost impossible to look for it beforehand. The mistake made was downplaying the bug to the average user. Computer architects have a responsibility to their customers to produce quality products. When a bug is discovered it should be standard practice to repair the problem without justification. These problems may not be a big deal to some users but to others it could mean years of research out the window or even human death.

Works Cited

- Pentium Study. 12 Dec. 1994. IBM. 12 Dec. 1994 `http://www.ibm.com/Features/pentium.html`. (This source was accessed through an archive titled Pentium Division Bug Documents available at `http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=1666&objectType=file`

- Peterson, Ivars. Pentium Bug Revisited. 12 May 1997. Mathematical Association of America. 27 Apr. 2005 `http://www.maa.org/mathland/mathland_5_12.html`.

- Wolfe, Alexander. "Intel Fixes A Pentium FPU Glitch." EE Times (1994)