

Forbes®

As Supreme Court Software Patent Ban Turns 40, It's

Time To Stop Ignoring It

By Timothy B. Lee – 11/21/12

On November 20, 1972, the Supreme Court handed down its first ruling on the patentability of software. In *Gottschalk v. Benson*, the Supreme Court invalidated a patent on a method for converting numbers from one binary format to another, “The mathematical formula involved here has no substantial practical application except in connection with a digital computer,” wrote Justice Douglas for a unanimous court. That, in his view, meant that the patent would “wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself.” Mathematical algorithms are not eligible for patent protection, so the Supreme Court invalidated the patent.

Of course, a similar argument could be made about any software patent. A computer program is nothing more than a sequence of mathematical operations—a complex mathematical formula. Therefore, any patent that claims a method of solving a problem by programming a general-purpose computer is, like the patent the high court struck down 40 years ago, effectively a patent on a mathematical algorithm.

Yet today the software industry is awash in patents and patent litigation. So what happened? Two things. First, the Supreme Court was not as clear as it could have been. In the same decision announcing the rule against patenting mathematical algorithms, the court also suggested that it might allow a patent that covers a “program servicing a computer.” What’s the difference between patenting an algorithm and patenting a “program servicing a computer?” The

Supreme Court didn't explain, and we're still scratching our collective heads four decades later.

But the more important step in the de facto legalization of software patents came in 1982, when Congress created a new court called the United States Court of Appeals for the Federal Circuit. It was given authority to hear all appeals in patent cases. While the new court was still theoretically under the authority of the Supreme Court, the fact that it heard so many patent cases gave it an outsized influence on patent law. And because the majority of its cases were patent cases, its judges spent a lot of time rubbing elbows with patent attorneys. As a result, it quickly shifted the law in the direction patent attorneys prefer: toward an ever stronger and more expansive patent system.

Beginning in 1989, the Federal Circuit began handing down a series of decisions that made it easier to get software patents. By the end of the 1990s, all practical limits to patents on software had been dismantled, sparking the software patent arms race that continues to this day.

Yet theoretically, the Supreme Court's 1972 ruling is still a binding precedent. The Supreme Court re-iterated its rule against patenting software in 1978. The Supreme Court did uphold a patent on a software-controlled rubber-curing machine in 1981, but its ruling emphasized that this was because the patent covered a physical machine that happened to have a software component, rather than claiming a software technique by itself.

Unfortunately, the Supreme Court hasn't made any effort to rein in the Federal Circuit on the software patent issue. While the Supreme Court saved us from patents on medical diagnostic techniques this year, it hasn't examined the validity of a software patent since 1981. It's past time for the Supreme Court to insist that lower courts respect its precedents, which, after all, are still the law of the land.