

# What does an NMI error mean? (The infamous "Hardware Malfunction")

 [devblogs.microsoft.com/oldnewthing/20070227-00](http://devblogs.microsoft.com/oldnewthing/20070227-00)

February 27, 2007



Raymond Chen

I promised to talk more about NMI, so here it is. What generates an NMI? What does it mean? The first question is easy to answer but doesn't actually shed much light: Any device can pull the NMI line, and that will generate a non-maskable interrupt. Back in the Windows 95 days, a few really cool people had taken the ball-point pen trick one step further: They had a special expansion card in their computer with a cord coming out the back. At the end of the cord was a momentary switch like the one you might see on a quiz show. If you pressed it, the card generated an NMI. No fumbling around with ball-point pens for these folks, no-ho! (To be honest, I had two of these. One of them was a simple NMI card, triggered by a foot pedal! The other was really a card with a high-resolution real-time clock that could be used for performance analysis. I used the NMI button far more often than the timer...) In practice, the only device that generates an NMI (on purpose) is the memory controller, which raises it when a parity error is detected. The non-geek explanation of a parity error: Your memory chips are acting flakey. Here's what a parity error looks like. It shows up as a mysterious "Hardware Malfunction" error. Now, it's possible that a device may be generating an NMI by mistake. For example, in Wendy's case, it may have been due to damaged caused by overheating. If you suspect your memory chips, you can run a memory diagnostic tool to see if it can find the bad memory.

My colleague Keith Moore reminded me that paradoxically, on the IBM PC-AT, you could mask the non-maskable interrupt! This definitely falls into the category of "Unclear on the concept." The masking was done in hardware that could be configured via some magic port I/O. It prevented the NMI from reaching the CPU in the first place. (NMI is still not maskable in the CPU.)

Raymond Chen

**Follow**

