



Wenn Daten-Bits umkippen

## Bug des Monats



**Ein Fall für die Müllabfuhr: Das betagte 386SX-Motherboard kommt beim DMA-Transfer ins Stolpern und wurde von CHIP ausgetauscht**

**E** begab sich zu jener Zeit, als Luigi Colani noch nicht seine Hände an die PC-Gehäuse der Firma Vobis gelegt hatte. Da entstand der Highscreen 386-SX-16 des CHIP-Lesers Tobias Knüttel. Doch weder das kantige PC-Gehäuse noch die mäßige Geschwindigkeit ist das Problem des Herrn Knüttel. Vielmehr schlägt er sich nach jedem Schreibzugriff des Diskettenlaufwerks mit der Fehlermeldung „Parity-Error... System halted“ herum. Danach steht der Rechner.

## Die Diagnose des CHIP-Test-Centers

**E**n Parity-Error tritt in der Regel auf, wenn ein RAM-Baustein defekt ist. Die meisten PC verfügen für jedes Byte im RAM über ein neuntes Bit, das Paritäts-Bit. Immer wenn ein Byte geschrieben wird, berechnet eine Logik den Wert dieses Bits und belegt es entsprechend. Der PC arbeitet mit einer sogenannten ungeraden Parität. Das bedeutet, daß die Summe aller neun Bits immer ungerade ist. Beim Auslesen eines Bytes überprüft wieder eine Logik diesen Wert. Stimmt das Ergebnis, passiert nichts; stimmt es nicht, wird dies der CPU über die NMI-Leitung mitgeteilt. NMI steht für Non Maskable Interrupt und hat eine Unterbrechung des Programmflusses zur Folge, die nicht gesperrt werden kann. Die korrespondierende BIOS-Routine sorgt dann für die genannte Fehlermeldung.

Ein einfacher Defekt im RAM war bei diesem Problemfall aber nicht wahrscheinlich, denn bei einem Diskettenzugriff wird der RAM-Speicher nicht sonderlich gestreßt.

Das lenkte meine Aufmerksamkeit in eine andere Richtung: Der Datentransfer zum Floppy-Laufwerk funktioniert aus historischen Gründen mittels DMA (Direct Memory Access). Der DMA-Controller übernimmt dabei die Kontrolle über den Bus und schickt „fly-by“ direkt Daten vom Memory zu einem I/O-Port oder umgekehrt. Die PC-Architektur benutzt für Disketten den 8-Bit-DMA-Kanal 2. Was wäre also, wenn bei dieser Übertragung etwas schiefginge?

Änderungen im (sehr kryptischen) CMOS-Setup brachten mich auch kaum weiter, mit der Ausnahme, daß nach dem Ausschalten des Interrupt-Modus für die Speicher-

bänke der Fehler verschwand – zusammen mit 4 der 8 Megabyte RAM. Der gleiche Effekt war durch den Ausbau von zwei der vier Speicherbänke zu erzielen.

Zusammenfassung: Immer dann, wenn zwei der vier RAM-Bänke aktiv sind und ein DMA-Transfer vom Speicher zu einem I/O-Port stattfindet, tritt ein Speicherfehler auf. Ich deute das so: Während des DMA-Transfers gerät der (übrigens auch vom DMA-Controller initiierte) Speicherrefresh ins Stocken. Bei der Bestückung mit zwei Bänken führt das noch nicht zu Verlusten, beim Vollausbau wird der Refresh-Zyklus zu lang. Das Resultat sind gekippte RAM-Zellen, was die Paritätslogik umgehend bemerkt. Ursache sind vermutlich ermüdete Treiberstufen des DMA- oder Memory-Controllers.



**Dipl.-Ing. Jan Kleinert, Redakteur im Ressort Ratgeber**

Der Beweis für diese Deutung ist schwer zu erbringen, da ein Logic-Analyzer – das ist ein Gerät zur Signalanalyse auf Bus-Systemen – auf das Kippen von RAM-Zellen nicht zu programmieren ist. Wirtschaftlich wäre die weitere Suche sowieso nicht, da der Wert des 386SX-Boards 20 Mark kaum übersteigt.

Ich ließ Herrn Knüttel trotzdem mit seinem Problem nicht allein: CHIP spendierte aus dem Fundus des Text-Centers ein 386DX40-Motherboard nebst passendem Speicher. Damit war der Fehler behoben und der Rechner sogar ein bißchen schneller.

Jan Kleinert