

Nicht nur der Pentium hat Fehler

Serielle Schnittstelle kritisch betrachtet

Franz Fiala

DSK-472: X00153A.ZIP, BNU202.ARJ, INT14.ZIP, CFOS.ZIP, STOMP.ZIP

Über den APPLE-IIe wurde berichtet, er sei in einer Autowerkstätte entstanden, eine Meldung, die seinen Erfolg nicht schmälerte, sondern ihn eher interessanter machte. Die Entstehungsgeschichte weist aber auf Bastelware hin, die sich dann doch auch im Betrieb durch das eine oder andere Symptom äußerte.

Vielleicht war es beim PC doch ein Entwicklungslabor, das als Geburtsstätte diente denn, gewisse Verbesserungen gegenüber dem APPLE weisen auf professionellere Eltern hin. Wenn aber einmal die wichtigsten Interrupts programmiert sind, geben die Techniker allzuleicht den Kaufleuten nach, die meinen, irgendwann müsse mit der Entwicklung Schluß sein. Und der serielle Interrupt ist einmal nicht lebenswichtig. Anders kann man kaum erklären, daß ein so wichtiger Punkt, die serielle Kommunikation so sparsam programmiert ist.

Auch die verwendeten UARTs vom Typ 8250 wiesen schon Jahrzehnte vor dem Pentium viel schwerwiegendere Fehler auf, denen zwar in den meisten Fällen durch geeignete Programmierung begegnet werden kann, aber immerhin zeigen diese Vorkommnisse, daß Fehler in ICs ziemlich häufig vorkommen, genauso wie Fehler in Programmen. Es ist nur die Frage, bis zu welcher Größenordnung man diese Fehler tolerieren kann und für welche es brauchbare Work-arounds gibt.

Aber der Reihe nach¹

Der 8250-Chip weist gleich eine ganze Reihe von Fehlern und Unzulänglichkeiten auf, die einen unvorbereiteten Programmierer ganz schön frustrieren können. Bekannte Probleme sind:

1. Ungültige Werte im Transmit Holding Register
2. Verlorene Interrupt Enable Bits
3. Beschränkte Baudrate
4. Probleme bei 8 Datenbits und gleichzeitig eingeschaltetem Paritätsbit

ad 1. Setzt man das *Interrupt Enable Register* bezüglich der Funktion *Byte senden*, dann generiert diese Funktion einen Interrupt, auch wenn das Senderegister noch gar nicht leer ist. Wenn diese Interruptfunktion verwendet wird, sollte der Interrupt nur verwendet werden, wenn das *Transmit Holding Register* leer ist.

ad 2. Diesen verlorenen Bits begegnet man durch zweimaliges Schreiben in das *Interrupt Enable Register* mit einer kurzen Zeitverzögerung. Kodestücke von dieser Art sind immer wieder im BIOS zu finden:

```
out dx,ax
jmp short $+2
out dx,ax
```

ad 3. Obwohl alle UARTS bis 115200 bit/s getaktet werden können, beschränkt der Erzeuger die Baudrate auf 56000 bit/s bei den Typen 8250B und 8250C. Die Serie 8250A hat diese Beschränkung nicht.

ad 4 Dieser Fehler kann nicht ausreichend reproduziert werden, vielleicht waren nur erste Serien dieser Chips betroffen.

Abhilfe

Nachfolger des 8250 sind die Bausteine 16450 und 16550. Beide haben einen eingebauten FIFO (First-In-First-Out-Speicher). Natürlich sind die oben genannten Fehler nicht mehr enthalten, dafür sind - wie könnte es anders sein - wieder neue Fehler, diesmal im FIFO, auf. Neuere Serien sind wieder fehlerfrei.

Das BIOS hilft

Welchen Chip immer Sie haben. Das BIOS versucht, diesen Fehlern zu begegnen und die Port-Ansprache durch Plausibilitätszugriffe fehlerfrei zu gestalten. Eine BIOS-Routine ist also dem direkten Chipregisterhandling vorzuziehen. Direkter Registerzugriff diene nur zu Schulungszwecken oder zur Entwicklung von Treibern für nicht unterstützte

Hardware. Der Vorteil, mit dem BIOS-Zugriff auch gleichzeitig besser portierbarere Software zu haben, ist inkludiert.

Kommunikation Programm-Ereignis

Der Geschwindigkeitsunterschied zwischen Hochsprachen- und Assemblerprogrammierung ist oft mehr als nur eine Größenordnung. Programme, die in Hochsprachen zu langsam sind, bekommen, in Assembler geschrieben, ungeahnte Geschwindigkeit. Genauere Analysen, wieviel Zeit Programme für bestimmte Abläufe benötigen, ermöglichen die Profiler. Aber auch wenn man die verbrauchte Zeit weiß, kann man sie in der Hochsprache zwar minimieren aber gewisse Laufzeiten nicht unterschreiten.

Die Kommunikation mit externen Ereignissen und einem Programm enthält das Problem, daß die beiden nicht synchronisiert sind, sodaß Abfragen im Programm, dieses Ereignis betreffend, manchmal zu langsam erfolgen und das externe Ereignis schon vorbei ist und daher die Erkennung durch das Programm verhindert wird. Üblicherweise bietet ein Interrupt eine kleine Hilfe. Die Hardware-Interruptroutine registriert das Ereignis und speichert es für das abholende Programm (beim PC ein Software-BIOS-Interrupt). Im PC gibt es 15 Hardware-Interrupts. Jedem Hardware-Interrupt stehen Software-Interrupts gegenüber. Für den Fall der seriellen Schnittstelle ist der Hardware-Interrupt für COM1 die Nummer 0Ch und für COM2 die Nummer 0Bh, der abholende Interrupt ist einheitlich der parametrisierbare Interrupt 14h.

Ungepufferter Interrupt im PC

Das abholende Programm ist eine Polling-Routine mit einer gewissen Laufzeit für den ungünstigsten Fall. Diese ungünstigste Zeitspanne lang wird ein ankommendes Byte in einem einzigen Byte zwischengespeichert. Dabei tritt dieser ungünstigste Fall durchaus nicht immer auf, er genügt aber schon, daß ein vorgegebenes Protokoll durch Auslassung eines Zeichens, denn dazu genau kommt es, nicht verifizierbar ist, die Übertragung kommt nicht zustande.

Abhilfe 1

Hat man Zeit, kann man sich mit Quittierungsprotokollen behelfen. Jedes empfangene Zeichen wird zum Sender zurückgeschickt, das nächste Zeichen wird erst nach Empfang der Quittung (und nicht nach dem leeren Sendepuffer) abgeschickt.

Abhilfe 2

Man kann auch kleine Pausen zwischen den gesendeten Zeichen an den richtigen Stellen, die dem ungünstigsten Empfangsfall nachgebildet sind einbauen. Die Folge, und auch gleichzeitig eine weitere Möglichkeit zu einer Milderung des Fehlers, ist eine Geschwindigkeitsreduktion.

Abhilfe 3

In moderneren Geräten sind von vornherein die Nachfolger des 8250, eben 16450 und 16550 eingebaut, die ein 16-stufiges Empfangsregister aufweisen und daher eine gewisse Elastizität gegenüber einer langsamen Empfangssoftware haben.

Diese kleinen Puffer genügen zwar, Programmlaufzeiten auszugleichen, nicht aber, um einen massiven Empfängerstopp, wie z.B. durch einen Plattenzugriff aufzufangen.

Abhilfe 4

Ein gepufferter serieller Interrupt würde - so wie bei der Tastatur - einen Stauraum für gesendete und empfangene Zeichen zur Verfügung stellen. Daß die Konstrukteure des PC-BIOS das nicht gewußt hätten, trifft nicht zu, denn bei der Tastatur ist ein solcher Puffer ja eingebaut. Man hat vielleicht der seriellen Schnittstelle damals, als noch ein Audio-Kassetten-Interface konzipiert war, die spätere Bedeutung nicht voraussehen können, spätestens beim AT wäre eine Möglichkeit zur Korrektur gegeben gewesen. Erst beim PS/2 von IBM und beim EISA- und PCI- Konzept wird dieser Mangel von Grundauf durch einen gepufferten seriellen Interrupt behoben. In PS/2-, EISA- oder PCI-Systemen tritt dieser Fehler nicht auf.

¹ Fehler des UART entnommen aus „The Undocumented PC“, Frank van Gilluwe, The Andrew Schulman Programming Series, ISBN 0-201-62277-7, § 44.95.

Abhilfe 5

Alle älteren XT/AT/386/486 müssen eine Software-Lösung benutzen, die aber durch sehr gute Shareware-Programme gut abgedeckt ist (wenn man's weiß). [Ein Grund mehr vielleicht, einmal in den FidoBoxen vorbeizuschauen, sie enthalten die eine oder andere wertvolle Software, die Sie schon lange gesucht haben. Ein Verzeichnis aller Boxen weiter vorne in diesem Heft macht den Zugang etwas leichter.] Programme, die jetzt dem ungepufferten Int14h des PC Beine machen, haben wir aus *His Master's Voice* in den Diskettendienst übernommen. Diese Programme werden FOSSIL-Treiber genannt, als Abkürzung für Fido-Opus-Seadog-Standard-Interface-Layer. Die beiden am häufigsten verwendeten Programme sind BNU und X. Genaugenommen sind INT14h-Programme Int14h+Bh+Ch-Programme, denn sie beeinflussen auch gleichzeitig die Hardware-Interrupts 0Ch und 0H. Da ein Anwendungsprogrammierer aber nie auf die Hardwareinterrupts zugreift, bleibt es bei der einfachen Bezeichnung Int14h.

Noch ein Plus für BIOS-Programmierer

Es stimmt, ein Registerzugriff ist das schnellste Betriebsmittel beim Umgang mit der Umwelt. Der BIOS-Zugriff ist aber der Schlüssel zu Programmen, auf die auch netzwerkweit zugegriffen werden kann. Sie heißen dann INT14h-fähig, da sie die gesamte Kommunikation über den Interrupt 14h abwickeln. Man kann daher mit solchen Programmen auf Modem-Ports vernetzter Rechner zugreifen, denn auch schon in kleineren Netzen ist es wenig sinnvoll, an jedem Rechner ein Modem zu installieren.

Drei Anwendungsbereiche werden damit abgedeckt:

- Modem-Kommunikation (Programm STOMP20.ZIP)
- ISDN-Kommunikation (Programm CFOS.ZIP)
- NASI/NACS-Ports in Novell-Netzen (Programm INT14.ZIP)

Das an anderer Stelle beschriebene Programm SUXXESS ist in diesem Sinne NASI- oder INT14h-kompatibel, für DOS-Anwender kann das Programm BTX-DELUXE empfohlen werden, das ebenfalls im Int14h-Sinne netzwerktauglich ist und aus dessen Lieferumfang die obigen Shareware-Programme entnommen sind. Weitere Hinweise zur Benutzung der FOSSIL-Software im Zusammenhang mit ISDN finden Sie in den **PCNEWS**-38, Seite 33. □

Programmieren lernen oder Listen abtippen

Barbara Haidner

„Wenn ich einen Computer bekomme, werde ich programmieren und viel Geld damit verdienen“. Die großen Pläne, mit der Programmierung origineller Spiele, viel Geld zu verdienen, setzen Mutter und Sohn einträchtig vor den ersten, teuer erworbenen Rechner. Die beigelegten Handbücher und der Basicemulator werden studiert und installiert. Zuerst tippen sie kurze, dann seitenlange Listen ab. Wie sollte man auch eigene Programme erstellen, wenn man nicht zuerst die Beispiele aus den Manuals ausprobiert? Kleine Fraktale und schöne, bunte Grafiken entstehen und warten auf die Weiterentwicklung. Bunte Magazine bieten jedoch immer neuere Listen an. Man kommt kaum mit dem Abschreiben der, die Augen strapazierenden Seiten nach. Doch verärgert stellt man fest, daß die Listen nicht halten, was sie versprechen. Jungprogrammierer bieten ihre Programme den Redaktionen an, die sie aus Zeitmangel nicht überprüfen können. So entsteht der erste Frust. Trotz Korrekturen in den nächsten Ausgaben funktioniert kaum eines, der groß angekündigten Spiele.

Eine diesbezügliche verärgerte Bemerkung der Mutter greift der Sohn freudig auf, läßt endlich Listen sein und kümmert sich wieder mehr um Schule und Freunde.

Plötzlich ergießt sich eine ganze Flut toller Games voller Raffinessen, Farbenvielfalt und mit tollem Sound über die User. Man vergißt total und gerne, daß man eigentlich selbst so etwas kreieren wollte. Die Programmierkenntnisse enden bei der Installationsroutine und einfachsten Commandlinebefehlen. Ob man nach der Entwicklung komplexester Applications, noch programmieren lernen soll, ist die Frage, die sich jeder selbst stellen muß.

Besonders jetzt, mit der Welle des Global village und Infohighways fühlt man schmerzhaft, daß man mit Programmierkenntnissen, dieses und jenes *Prog* besser an seine Bedürfnisse anpassen könnte. FIDO- und andere Netze fordern etwas mehr als die übliche Beantwortung von Dialogabfragen zum Installieren eines Point-Mailbox- oder Terminalprogramms. Bei der Vielfalt der Programmiersprachen kein leichtes Unterfangen. Wo und wie kann man lernen, ohne überflüssigen, theoretischen Ballast? Gefragt ist die Praxis und Praktiker, die genügend Zeit und Geduld für Newcomer haben, sie nicht mit Fachausdrücken abschrecken. □

Intels Top 10-Ausreden zum Pentium-Divisionsfehler

- Platz 10:** Eigentlich sollten wir den Preis erhöhen, schließlich haben wir umsonst einen Zufallsgenerator eingebaut.
- Platz 9:** Versuch mal DAS hier zu emulieren, Power PC!
- Platz 8:** Wir sind jetzt im Guinness-Buch der Weltrekorde: Teuerster Briefbeschwerer der Welt
- Platz 7:** Hauptsache die High-Score-Berechnung in Tetris ist einwandfrei
- Platz 6:** He, haben Sie schon mal 3 Millionen winzige Transistoren beim ersten Mal richtig miteinander verbunden?
- Platz 5:** Das ist Teil unserer neuen "Fuzzy Logic"-Produktstrategie.
- Platz 4:** Cyrix und AMD sind jetzt garantiert nicht 100 Prozent KOMPATIBEL!
- Platz 3:** Wie, 2 plus 2 ist nicht 3,99999999998456?
- Platz 2:** Was denken Sie denn, warum das FLIESSKOMMA heißt?
- Platz 1:** Endlich haben Sie eine gute Ausrede bei Fehlern in der Einkommenssteuererklärung!

TOP TEN NEW INTEL SLOGANS FOR THE PENTIUM

- 9.9999973251 It's a FLAW, Dammit, not a Bug
- 8.9999163362 It's Close Enough, We Say So
- 7.9999414610 Nearly 300 Correct Opcodes
- 6.9999831538 You Don't Need to Know What's Inside
- 5.9999835137 Redefining the PC--and Mathematics As Well
- 4.999999021 We Fixed It, Really
- 3.9998245917 Division Considered Harmful
- 2.9991523619 Why Do You Think They Call It *Floating* Point?
- 1.9999103517 We're Looking for a Few Good Flaws
- 0.9999999998 The Errata Inside

- P** Produces
- E** Erroneous
- N** Numbers
- T** Through
- I** Incorrect
- U** Understanding of
- M** Mathematics

gefunden im Internet von Fritz Zetik