

Prof. Dr. Christoph Scholl
Dr. Tim Welschehold
Alexander Konrad
Niklas Wetzel

Freiburg, 11.11.2022

Betriebssysteme

Übungsblatt 4

Aufgabe 1 (3+4 Punkte)

Gegeben sei ein Rechner mit einem Prozessor, der mit einer Taktrate von 800 MHz arbeitet. Es besteht einerseits die Möglichkeit, die Kommunikation mit der Festplatte Interrupt-getrieben abzuwickeln und andererseits die Möglichkeit, DMA (= Direct Memory Access) zu verwenden. Die Festplatte hat eine Datenübertragungsrate von 8MB/sec.

- a) Nehmen Sie für die Interrupt-getriebene Kommunikation an, dass die Festplatte jeweils acht 32-Bit-Worte auf einmal zum Prozessor überträgt und dass sie dem Prozessor jeweils durch Auslösen eines Interrupts mitteilt, dass die Daten bereit sind. Der Overhead, der bei dem Prozessor für eine Datenübertragung (einschließlich Interrupt) anfällt, betrage 1000 Taktzyklen. Nehmen Sie weiterhin an, dass die Festplatte nur während 5% der Zeit überhaupt aktiv ist. Wie groß ist in diesem Fall der relative Anteil der CPU-Zeit, der für Datentransfers von Festplatte zum Prozessor aufgewendet wird?
- b) Nehmen Sie für den DMA-Transfer an, dass jeweils 16KB-Blöcke auf einmal von der Festplatte zum Speicher übertragen werden. Die Übertragung wird begonnen mit einer Aktivierung des DMA-Controllers durch den Prozessor. Danach überträgt die Festplatte gesteuert vom DMA-Controller Daten direkt zum Speicher. Nehmen Sie der Einfachheit halber an, dass während dieser Zeit bei der Arbeit des Prozessors keine Konflikte auf dem Systembus entstehen, so dass der Prozessor ohne Beeinträchtigung durch den DMA-Transfer weiterarbeiten kann. Die Beendigung des Transfers meldet der DMA-Controller dem Prozessor über einen Interrupt. Der Prozessor benötige 1500 Taktzyklen, um den DMA-Controller vor einer Übertragung eines Blockes zu aktivieren. Außerdem benötige der Prozessor weitere 500 Taktzyklen zur Bearbeitung des Interrupts, der vom DMA-Controller nach Beendigung der Blockübertragung ausgelöst wird. Berechnen Sie auch hier den relativen Anteil der CPU-Zeit, den der Prozessor für Datentransfers aufwendet, *unter der Annahme, dass die Festplatte ebenfalls nur während 5% der Zeit überhaupt aktiv ist.*

Hinweis: Bitte geben Sie Ihren Rechenweg an!

Aufgabe 2 (3+4 Punkte)

Betrachten Sie einen Prozessor mit nur einem Interruptlevel, der zur Verarbeitung von Interrupts mit verschiedenen Prioritäten mit einem Interrupt-Controller zusammenarbeitet. Wenn auf dem Prozessor keine Interrupt-Service-Routine (ISR) aktiv ist oder wenn ein am Interrupt-Controller anliegender Interrupt eine höhere Priorität hat als die aktuell auf dem Prozessor laufende ISR, dann signalisiert der Interrupt-Controller dem Prozessor über ein Signal INT , dass ein Interrupt anliegt, der den Prozessor unterbrechen darf. Nach Abarbeiten einer ISR signalisiert der Prozessor dem Interrupt-Controller durch Aktivieren eines Signals $/INTA$ für genau einen Takt, dass die ISR beendet wurde. Wir nehmen an, dass es maximal 255 (Hardware-)Interrupts mit Prioritäten 0 bis 254 gibt. Solange ein Interrupt INT_j noch nicht verarbeitet ist, darf das entsprechende I/O-Gerät j keinen weiteren Interrupt auslösen. Prozessorinterne Exceptions können Sie bei dieser Aufgabe vernachlässigen.

- a) Überlegen Sie sich eine Methode, wie der Interrupt-Controller feststellen kann, dass auf dem Prozessor aktuell gerade keine ISR läuft, die ihm vom Interrupt-Controller mitgeteilt wurde. Sie können dazu einen 8-Bit-Zähler verwenden, der über zwei Signale **up** und **down** gesteuert wird. Wenn **up** = 1, **down** = 0, dann zählt der Zähler bei steigender Clockflanke hoch, wenn **up** = 0, **down** = 1, dann zählt der Zähler bei steigender Clockflanke runter. Durch Aktivieren des Eingangs **/reset** kann der Zähler auf 0 initialisiert werden. Es genügt, wenn Sie die Methode mit Worten skizzieren.
- b) Reicht Ihre in Teil a) entwickelte Methode aus, um den Interrupt-Controller korrekt zu implementieren? Wenn nein, skizzieren Sie eine Situation, in der der Interrupt-Controller keine ausreichenden Informationen hat, um zu entscheiden, ob er über INT einen Interrupt signalisieren darf oder nicht. Entwerfen Sie nun eine Lösung, bei der der Interrupt-Controller das Signal INT immer korrekt setzt. Sie können annehmen, dass der Interrupt-Controller intern einen kleinen Speicherbereich mit 256 Speicherzellen hat, der über 8-Bit-Adressen angesprochen werden kann. Die Speicherzellen haben eine Wortbreite von 8 Bit. Weiterhin können Sie den 8-Bit-Zähler aus Teil a) in geeigneter Weise verwenden. Sie können außerdem annehmen, dass der Interrupt-Controller die Nummer des anliegenden Interrupts mit der höchsten Priorität stets in einem Register IVN speichert und zusätzlich dessen Priorität in einem Register PR speichert.

