



Übungen zur Vorlesung  
„Technische Informatik II“, WS 2002/03

{hauck,schmied,guenkova,prager}@vs.informatik.uni-ulm.de

Lösungsskizze Übungsblatt 6 - „Speicherverwaltung und E/A-Geräte“

### Aufgabe 1: Speicherverwaltung

- a) Segmentierung ist eine logische Unterteilung des gesamten Adressraums in (sich ggf. auch überlappende) Speicherbereiche. Diese können über Parameter in Segment-Deskriptoren zum Speicherschutz herangezogen werden. Beispielsweise können Segmente als schreibgeschützt, b.z. explizit als ausführbar gekennzeichnet werden. Paging bezeichnet das heute verbreitete Verfahren zur Realisierung virtuellen Speichers. Der logische Adressraum wird dabei in 4kB (bzw. 4MB) grosse Kacheln (Pages) unterteilt. Im laufenden Betrieb werden die gerade benötigten Kacheln mit Hilfe der MMU (Memory Management Unit) auf den (normalerweise kleineren) physikalischen Speicher abgebildet. (Weiteres siehe Vorlesungsfolien)
- b)
1. Zugriff auf die Speicheradresse geht von CPU an Cache
  2. Cache findet das Datum in keiner Cacheline („Cache Miss“)
  3. Cache Miss veranlasst die MMU die Daten neu in den Cache einzulagern. Wir gehen hier davon aus, dass die betrachtete Cache-Stufe zwischen CPU und MMU liegt und nicht direkt auf den Speicher zugreift. Bei mehreren Caches (L1, L2, ...) muss man unterscheiden, wie der jeweilige Cache mit MMU und Speicher arbeitet.
  4. im TLB (Translation Lookaside Buffer) könnte momentan die Abbildungsinformation für die Page, auf der die gesuchten Daten liegen, fehlen.
  5. die MMU muss also über den dazugehörigen Page-Table-Eintrag die logische zu einer physikalischen Adresse auflösen
  6. Zur Adressauflösung wird über das permanent im Speicher befindliche Page Directory eine Page Table gesucht, die die Daten-Page beschreibt (wohin abgebildet, present, dirty, ...). Diese beschreibende Page Table liegt natürlich auch im Speicher auf irgendeiner Page, die ebenfalls momentan ausgelagert sein könnte *Rightarrow* es wird ein Page Fault erzeugt
  7. der Page Fault wird dem Betriebssystem via Interrupt gemeldet
  8. die Page Fault-Routine des OS lädt die Page der Page-Table in den Speicher
  9. nun wird nochmals versucht auf die Daten-Page zuzugreifen. Es handelt sich hier um eine komplette Wiederholung des Vorgangs (zurück zu 1. !!).
  10. die Adressauflösung mittels Page Table funktioniert jetzt, jedoch ist die Daten-Page auch nicht im Speicher (present nicht gesetzt) *Rightarrow* es wird ein weiterer Page Fault erzeugt (*rightarrow*Interrupt*rightarrow*
  11. das Betriebssystem lagert die Datenseite ein und setzt u.a. das present-Flag im Page-Table-Eintrag.
  12. der Speicherzugriff (inkl. Eintrag im TLB und im Cache) kann nun erfolgen. Dazu wird wieder der ganze Vorgang (von 1. an !!) wiederholt.

Anmerkung: „Vorgang wiederholen“ heisst, dass der Opcode, der den Fault ausgelöst hatte, nochmal ausgeführt wird.

- c) Welche Komponenten (Hard- und Software!) sind hier involviert?  
 Hardware: CPU, Cache, MMU, TLB, RAM  
 Software: Page Directory/Table/Descriptor, Interrupt-Handler (, Festplatten-I/O)

## Aufgabe 2: Serielle Schnittstelle

- a) Leitungen: Receive RxD, Transmit TxD, Masse GND, Pegel: logisch 1 = -12 V, logisch 0 = +12 V
- b) Vorgehen: Länge des kürzesten Signals (Informationsträger) suchen. Damit findet man einen Hinweis auf die wahrscheinliche (momentan erkennbare) untere Grenze der Übertragungsgeschwindigkeit. Im folgenden skizziert für das erste Signal:  
 Kürzestes Signal ist  $104\mu s$  lang  $\Rightarrow \frac{1\text{Bit}}{0.000104} s \sim 9,6 \frac{kB}{s}$  minimale Übertragungsgeschwindigkeit. Ausgehend vom Ruhezustand (1, -12V) beginnt das Signal mit einem Startbit (0) und würde dann die Bitfolge 1, 0, 1, 1, 1, 1, 1, 1 repräsentieren, gefolgt von einem Stoppbit (nicht direkt erkennbar, da auch „1“). Da das Least Significant Bit zuerst gesendet wird erkennt man den Wert  $(11111101)_2 = 253$ . Man könnte nun diverse übliche Übertragungsgeschwindigkeit durchprüfen, z.B. 14,4 kB/s, 28,8 kB/s. Wir testen als nächstes den doppelten des ersten Wertes: 19,2 kB/s.  
 Wir erhöhen dazu die „Abtastgeschwindigkeit“ und betrachten eine Signaldauer von  $52\mu s$ . Nach dem Startbit lesen wir die Wertfolge 0, 1, 1, 0, 0, 1, 1, 1 (+ Stoppbit 1) ab mit dem Binärwert  $(11100110)_2 = 230$ .  
 Bei einer weiteren Verdopplung der Geschwindigkeit mit einer Signaldauer von  $26\mu s$  erkennt man das eine Übertragungsgeschwindigkeit von 38,4 kB/s nicht gewählt worden sein kann. Die abgelesene Wertfolge lautet: (Startbit 0, ) 0, 0, 0, 1, 1, 1, 1, 0. Jetzt müsste eigentlich ein Stoppbit mit Wert „1“ folgen, der Pegel bleibt aber weiterhin auf „0“.  
 Um die maximale Übertragungsgeschwindigkeit abzuschätzen kann man das längste Signal mit Wert 0 suchen. Grund: Spätestens nach einer 8-bit Null muss eine Stoppbit-Flanke nach 1 folgen. Im vorliegenden Signal könnten also auch zwei Nullen übertragen worden sein, getrennt von einer „8-bit“ Pause. Die Signaldauer beträgt in diesem Fall  $\frac{104\mu s}{1(\text{Startbit})+8(\text{Datenbits})} = 11,5\mu s$  mit einer Übertragungsgeschwindigkeit von  $\frac{9\text{Bit}}{0.000104} s \sim 86,5 \frac{kB}{s}$ , die Pause dauert entsprechend  $92,45\mu s$ .

## Aufgabe 3: Hypothetischer Bus

Sie sollen für ein Labor eine billige Busschnittstelle für maximal vier handgefertigte I/O-Steckkarten fertigen (es gibt also genau vier Busslots). Kurioserweise steht ein 8 Bit Microcontroller (8 Bit Adressbus, 8 Bit Datenbus) zur Verfügung, der Speicher wird aber mit 16 Bit adressiert ( $64kB = 2^{16} kB$ ) und die Steckkarten verwenden einen 10 Bit Adressraum ( $1024 \text{ Byte} = 2^{10}$ ). Die CPU kann also nur mit 8 Bit auf den Speicher zugreifen. Beim Zugriff auf die Steckkarten ist sie noch mehr eingeschränkt: 2 Bit der 8 Bit Adresse werden als Chip Select (CS) verwendet und via Decoder auf die vier Slots verteilt. Je nach angelegtem Wert  $(00)_2$  bis  $(11)_2$  wird eine der vier Chipselect-Eingänge mit „1“ beschaltet.

- a) Man benötigt zumindest eine IRQ-, eine IACK-Leitung, 8-Daten/Adress- Leitungen (CPU-Bus), eine Leitungen zur Auswahl des Adressziels (ist RAM adressiert oder Hardwareadressen auf den Steckkarten) und 2 Leitungen (2bit = Werte 00, 01, 10, 11) zur Adressierung der vier Slots (die hier in zwei Adressleitungen aufgehen).

