

Musterlösung Blatt 13: Busse und Ein- Ausgabe

1 Aufgabe 1: Datenübertragung

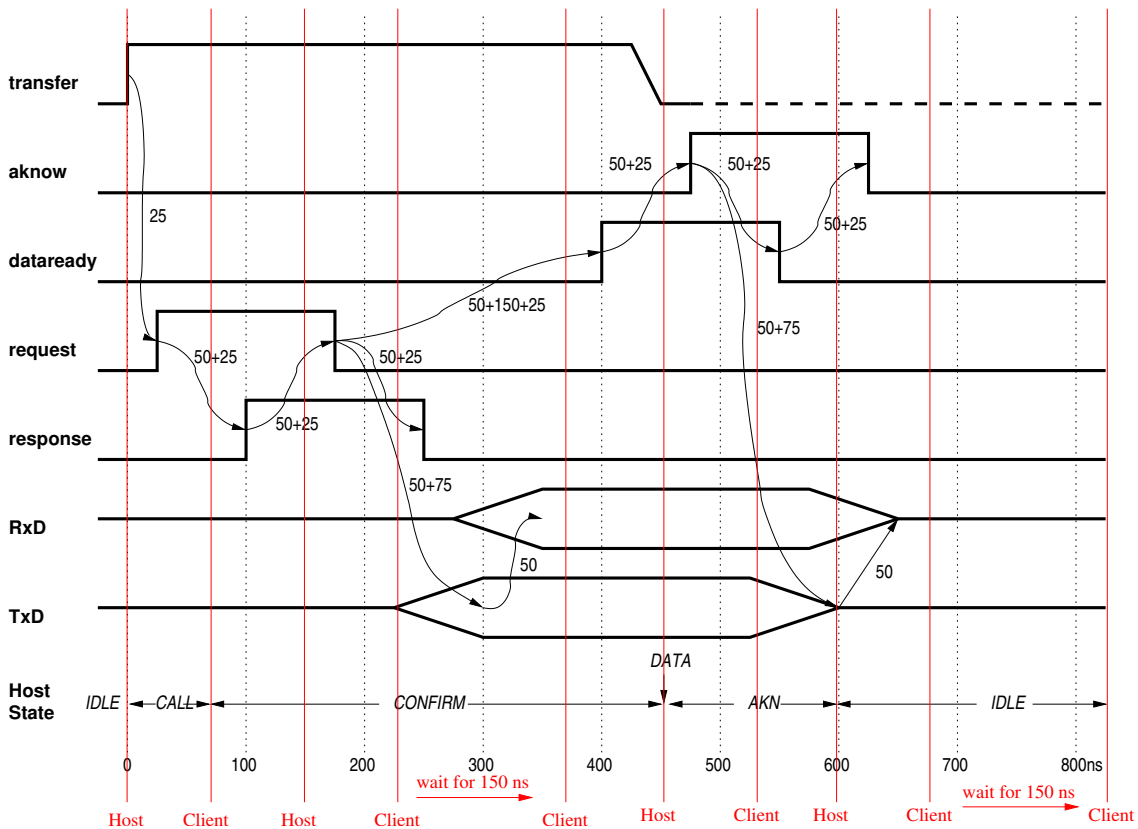


Abbildung 1: Timing des Protokolls (beobachtet jeweils auf der Sende-Seite der Signale). Es handelt sich hierbei um ein *half-interlocked* Protokoll (der Client ist zeitgesteuert, man beachte insbesondere die `wait for 150ns`; Anweisungen). Die roten Linien kennzeichnen die Zeitpunkte an denen einer der Teilnehmer auf ein Signal der Gegenseite oder das ablaufen eines Timers reagiert. An diesen Stellen werden Codezeilen ausgeführt die eine Wirkung (u.U. durch "after" verzögert) auf die Signale haben.

2 Aufgabe 2: Memory-PCI-Bridge

1. Wie lang ist ein Takt auf dem Memory-Bus?

$$T_{MEM} = \frac{1}{f_{MEM}} = \frac{1}{100 \cdot 10^6 \frac{1}{s}} = 10\text{ns}$$

2. Wie lang ist ein Takt auf dem PCI-Bus?

$$T_{PCI} = \frac{1}{f_{PCI}} = \frac{1}{66 \cdot 10^6 \frac{1}{s}} \approx \frac{1}{\frac{2}{3} \cdot 100 \cdot 10^6 \frac{1}{s}} = 15\text{ns}$$

3. Wieviel Memory-Takte ist ein PCI-Takt lang?

Einführung von Pseudoeinheiten " C_{MEM} " und " C_{PCI} " für "Memory-Bus-Takte" und "PCI-Bus-Takte".

$$k = \frac{T_{PCI}}{T_{MEM}} = \frac{15\text{ns}}{10\text{ns}} = 1,5 \frac{C_{MEM}}{C_{PCI}}$$

$$C_{PCI} = \frac{3}{2} C_{MEM}$$

4. Wieviel PCI-Takte ist ein Memory-Takt lang?

$$C_{MEM} = \frac{2}{3} C_{PCI}$$

5. Zeichnen Sie die beiden Takte für 6 Memory-Takte untereinander!

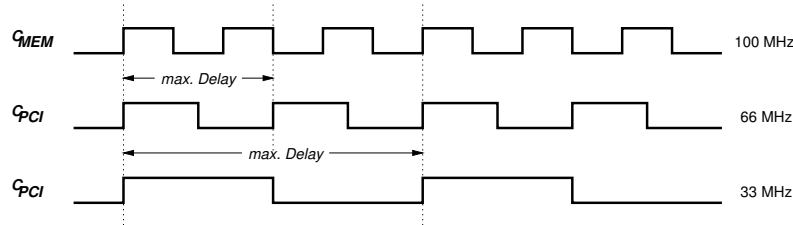


Abbildung 2: Memory- und PCI-Takt. Synchronisation kann stattfinden, wenn die Flanken beider Takte zur gleichen Zeit auftreten.

6. Wieviel Memory-Bustakte ist maximal die Verzögerung, bis die Transaktion auf dem PCI-Bus beginnen kann?

Geht man davon aus, dass selbst wenn beide Takte gleichzeitig ansteigen, der Durchgang durch die PCI-Bridge verzögert ist, so muss man bis zum nächsten positiven Flankewechsel warten. Somit ist die Verzögerung $D_{max} = 1,5 C_{MEM}$

7. Wieviel Memory-Bustakte dauert die Transaktion, wenn die Transaktion innerhalb eines PCI-Taktes abgeschlossen wird?

Nach dem Beginn einer Transaktion kann die nächste immer nur bei der nächst folgenden positiven Flanke erfolgen. Somit verlängert sich die Zeit bis zu dieser (Annahme: keine Verzögerung beim Durchgang durch die Bridge).

$$D_T = \lceil C_{PCI} \cdot k \rceil = \lceil 1,5 C_{MEM} \rceil = 2 C_{MEM}$$

8. Wie lange dauert die gesamte Transaktion insgesamt im schlechtesten Fall?

$$D(1) = \lceil C_{PCI} \cdot k + D_{max} \rceil = \lceil 1,5 C_{MEM} + 1,5 C_{MEM} \rceil = 3 C_{MEM}$$

9. Wie lange dauert die gesamte Transaktion insgesamt, wenn die Transaktion n Takte auf dem Memory-Bus benötigt?

$$D(n) = \lceil n \cdot C_{PCI} \cdot k + D_{max} \rceil = \lceil n \cdot C_{PCI} \cdot k + 1,5 C_{MEM} \rceil$$

10. Wie hoch ist die reine Netto-Datenrate (Byte pro Sekunde), die auf dem PCI-Bus erreicht werden kann? (In der Literatur wird oft nur die Brutto-Rate mit $266,6 \frac{\text{MByte}}{\text{s}}$ angegeben.)

$$66 \cdot 10^6 \text{Hz} \cdot 32 \text{Bit} \cdot \frac{1}{2} \cdot \frac{1}{8} \frac{\text{Byte}}{\text{Bit}} = 132 \cdot 10^6 \frac{\text{Byte}}{\text{s}} = 132 \frac{\text{MByte}}{\text{s}}$$

Hierbei beachte man, dass auf den 32 Datenleitungen des PCI-Busses auch die Adressen mit übertragen werden müssen. Es handelt sich um einen sogenannten *Multiplexed-Bus*. Für eine reine Netto-Datenübertragung sind also 2 Zyklen auf dem PCI-Bus (für Adresse und Daten) nötig (deshalb der Faktor $\frac{1}{2}$).

3 Aufgabe 3: Interrupt-Behandlung mit Daisy-Chain

Bis die CPU das $\overline{\text{INT}}$ -Signal akzeptiert, muss eventuell ein Befehl noch vollständig abgearbeitet werden. (Ein Interrupt kann nur am Anfang eines vonNeumann-Zyklus beginnen.) Bis das Signal **Aknow** von der CPU aktiviert wird, können somit maximal bis zu 333ns vergehen.

Bis das Signal **Aknow** am I/O-Device 3 angekommen ist, vergehen aufgrund der Verzögerungen in der Daisy-Chain nochmal $2 \times 133\text{ns} = 266\text{ns}$.

Dann dauert es nochmal $2 \times 100\text{ns}$ bis das I/O-Device 3 seine Adresse über die Datenleitungen der CPU mitgeteilt und die CPU diese gelesen hat. Erst jetzt weiss die CPU, welches Device den Interrupt ausgelöst hat.

Jetzt muss die CPU noch die Sprungadresse des ersten Befehls der ISR aus der Vektortabelle in den *Programm Counter* (PC) laden. Das dauert nochmal 50ns.

Es vergehen also insgesamt bis zu maximal $333\text{ns} + 2 \times 133\text{ns} + 2 \times 100\text{ns} + 50\text{ns} = 849\text{ns}$, bis die CPU mit der Bearbeitung des Interrupts beginnen kann.