

# Rechnerstrukturen

Wintersemester 2003/2004

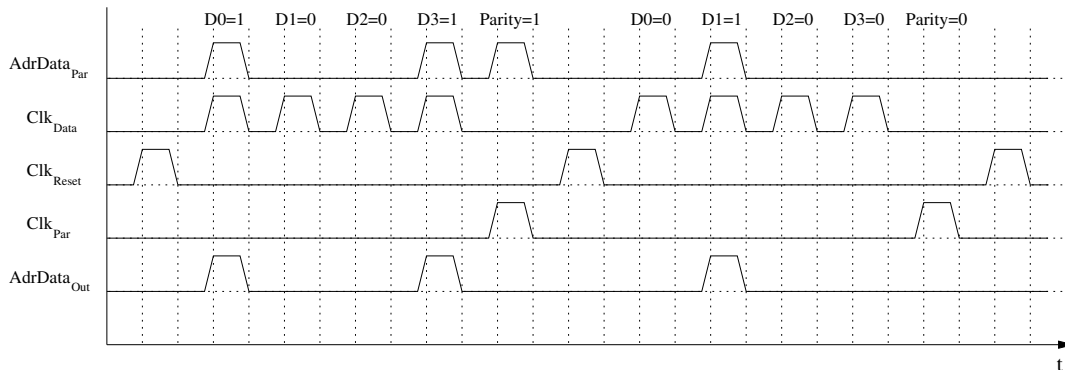
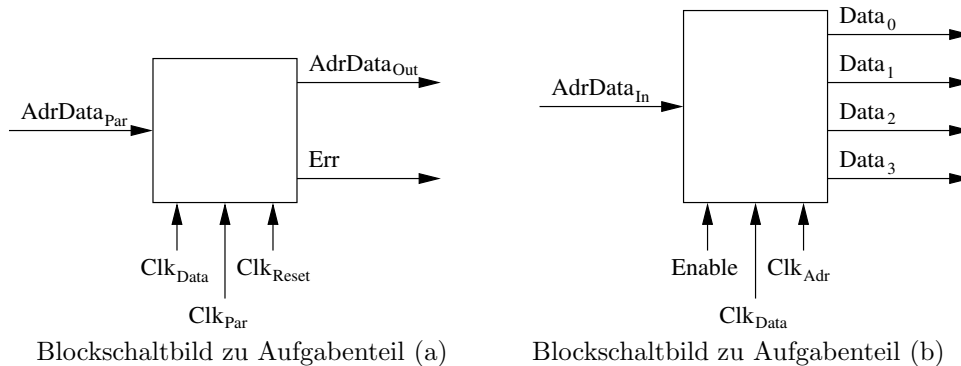
## 6. Übungsblatt

Besprechungstermin: 06.01.04, 8:15 Uhr in HZ 13

### Aufgabe 1:

Es soll eine gemultiplexte Datenübertragung von vier Datenströmen realisiert werden. D.h. unter Benutzung einer Leitung sollen vier Datenströme in je einem virtuellen Kanal übertragen werden. Zur Selektion des Kanals dient eine 2-Bit-Adresse, die zusammen mit den Daten auf der gleichen Leitung übertragen wird. Auf der Senderseite wird das gemischte Daten-/Adreßsignal inklusive ungeradem Paritätsbit erzeugt. In dieser Aufgabe soll die Logik für die Empfängerseite entwickelt werden. Als Eingabesignale stehen Ihnen das Daten-/Adreßsignal auf der Leitung  $AdrData_{Par}$  sowie mehrere Taktsignale nach untenstehendem Beispiel zur Verfügung. Realisieren Sie die Empfängerlogik in folgenden Schritten:

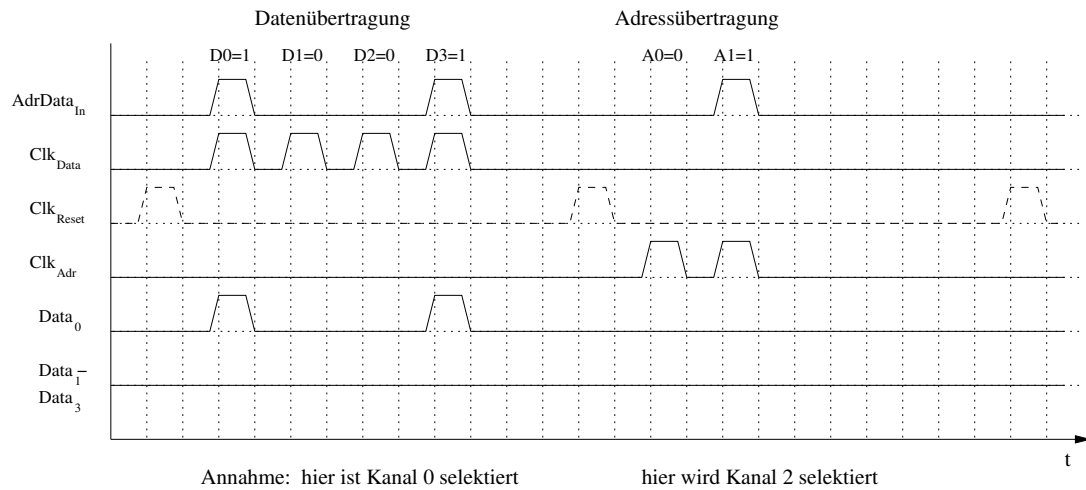
- Überprüfen Sie das Daten-/Adreßsignal  $AdrData_{Par}$  auf korrekte Übertragung, indem Sie aus den ersten vier Bits das erwartete (ungerade) Paritätsbit generieren und dieses mit dem aus dem empfangenen Signal extrahierten Paritätsbit vergleichen. Im Fehlerfall soll ein Ausgangssignal  $Err$  erzeugt werden. Dieses Signal muß lediglich zum Zeitpunkt der Paritätsprüfung gültig sein. Dabei wird eine fehlerhafte Übertragung durch  $Err = 1$  gekennzeichnet. Entfernen Sie das Paritätsbit aus dem Datenstrom, so daß dieser am Ausgang  $AdrData_{Out}$  zur Verfügung steht.



Als Beispiel werden die Datenblöcke 1001 und 0100 verarbeitet.

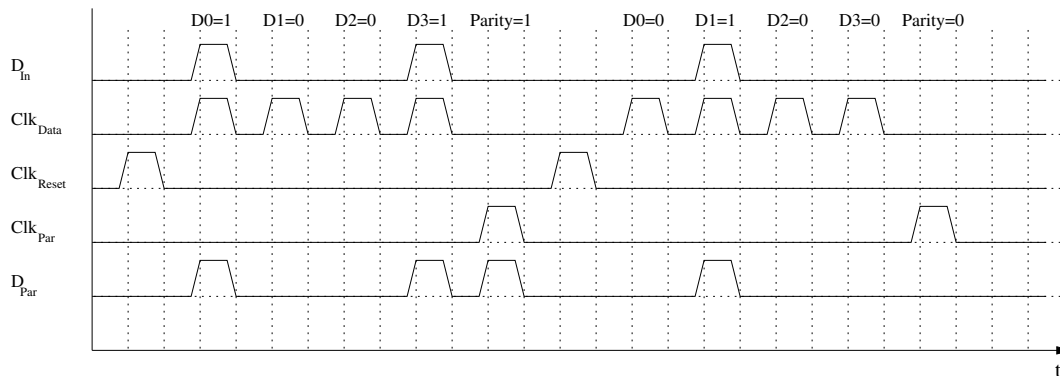
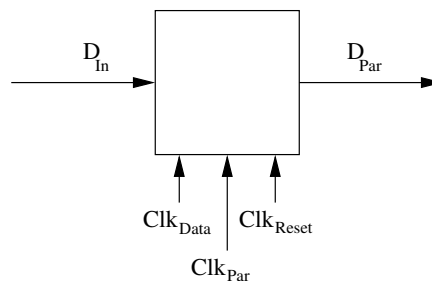
- Entwerfen Sie ein Schaltung, die den Datenstrom entsprechend der übertragenen Kanaladresse auf einen von vier Ausgangsleitungen umlenkt. Die Übertragung einer Adresse kann anhand des entsprechenden Taktsignals  $Clk_{Adr}$  erkannt werden. Die Schaltung soll einen Eingang  $AdrData_{In}$  besitzen, der das Daten-/Adreßsignal ohne Paritätsbit enthält. Die vier Ausgangsleitungen werden als  $Data_0..Data_3$  bezeichnet. Zusätzlich soll ein Eingang  $Enable$  bereitgestellt werden, der die Weitergabe des Eingabestromes auf die Ausgangsleitungen steuert. Bei  $Enable = 1$  werden die Daten auf den selektierten Ausgangskanal umgelenkt, während bei  $Enable = 0$  alle Ausgangskanäle auf log. 0 liegen sollen.

Als Beispiel sei aktuell der Kanal 0 selektiert. Daraufhin erfolgt eine Datenübertragung gefolgt von einer Adreßübertragung, die Kanal 2 selektiert. Im Beispiel wird implizit davon ausgegangen, daß während der Adreßübertragung der *Enable*-Eingang auf log. 0 liegt.



### Aufgabe 2:

Es ist ein Paritätsgenerator für einen seriellen Datenstrom zu entwerfen. Nach Übertragung jedes Datenblocks soll ein Paritätsbit für ungerade Parität an den Datenstrom angehängt werden. Dieses Paritätsbit dient letztendlich dazu, den Datenblock derart um ein Bit zu erweitern, dass die Gesamtzahl 1-en ungerade ist. Jeder eingehende Datenblock besteht hierbei aus vier Bits, die auf einer Leitung  $D_{In}$  bereitgestellt werden. Ferner stehen Ihnen drei Taktsignale  $Clk_{Data}$ ,  $Clk_{Par}$  und  $Clk_{Reset}$  nach untenstehendem Beispiel zur Verfügung. Während der  $Clk_{Data}$ -Impulse stehen die Datenbits in der Reihenfolge  $D_0$ ,  $D_1$ ,  $D_2$  und  $D_3$  am Eingang  $D_{In}$  bereit. Sie können davon ausgehen, dass diese Signale inkl. benötigter Setup- bzw. Hold- Zeiten stabil bleibt. Am Ausgang  $D_{Par}$  der Schaltung soll der (während des Taktimpulses  $Clk_{Par}$ ) um das Paritätsbit erweiterte Datenblock bereitgestellt werden. Somit ergibt sich für die zu realisierende Schaltung folgendes Blockschaltbild:



Als Beispiel werden die Datenblöcke 1001 und 0010 verarbeitet.