

# 5. Übung

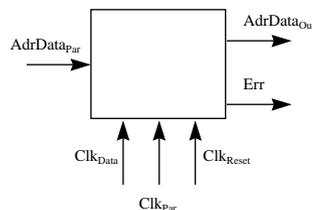
## Vorlesung Rechnerstrukturen WS 99/2000

---

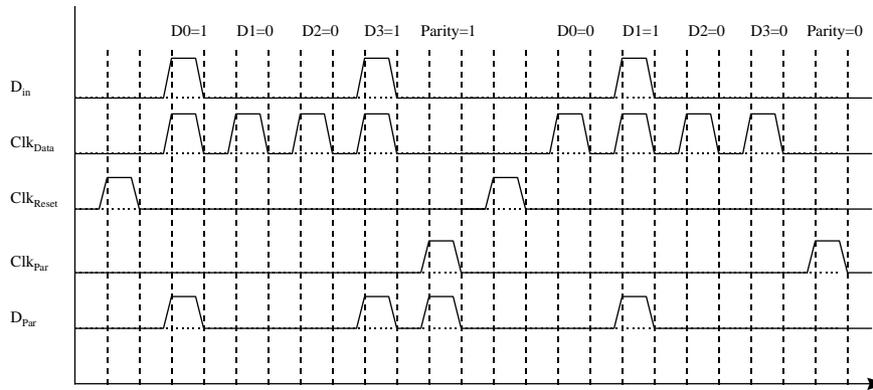
1. Erweitern Sie das JK-Master/Slave-Flip-Flop aus der Vorlesung um einen asynchronen Reset-Eingang. Der Ausgang Q des Flip-Flops soll durch Anlegen einer log. 1 am Reset-Eingang unmittelbar auf log. 0 gehen.
2. Realisieren Sie einen 8-Bit Addierer mit seriellen Eingängen und parallelem Ausgang. Verwenden Sie dabei *einen* Volladdierer sowie weitere aus der Vorlesung bekannten Bausteinen Ihrer Wahl. An den (seriellen) Eingängen A und B wird mit jedem Clock-Impuls ein Bit der Summanden bereitgestellt. Die Ausgabesignale  $S_0..S_7$  sowie  $Carry_{Out}$  werden erst nach acht Taktimpulsen, d.h. nach Verarbeitung aller acht Summandenbits als gültig betrachtet.
3. 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

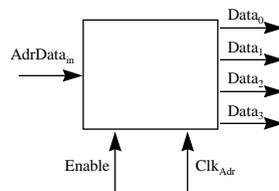
- 3.a. Überprüfen Sie das Daten-/Adreßsignal  $AdrData_{par}$  auf korrekte Übertragung, indem Sie aus den ersten vier Bits das erwartete (ungerade) Paritätsbit generieren und dieses mit dem aus dem empfangenen Signal extrahierten Paritätsbit vergleichen. Im Fehlerfall soll ein Ausgangssignal Err erzeugt 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. Somit ergibt sich für die zu realisierende Schaltung folgendes Blockschaltbild:



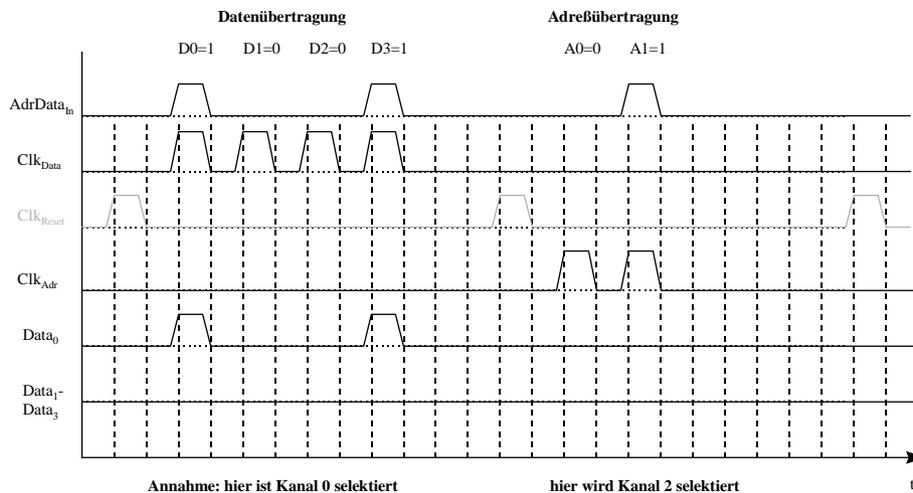
Als Beispiel werden die Datenblöcke 1001 und 0100 verarbeitet



3.b. 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. Somit ergibt sich für die zu realisierende Schaltung folgendes Blockschaltbild:



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.



Ausgegeben: 3.12.99

Abgabe: bis spätestens Freitag 10.12.99 vor der Vorlesung