

# 8. Übung

## Vorlesung Rechnerstrukturen WS 1999/2000

---

Es soll ein Funktionsbaustein entworfen werden, der einen eingehenden seriellen Datenstrom um ein Paritätsbit erweitern und an eine wählbare Menge von Ausgangskanälen weiterleiten kann. Der Baustein soll dabei durch folgende Kommandos konfiguriert werden können: 20 P

- *Wahl des Paritätsbits*  
Entscheidet, ob der eingehende Datenstrom um ein gerades, ein ungerades oder kein Paritätsbit erweitert werden soll.
- *Selektion der Ausgangskanäle*  
Die Ausgabe des —ggf. um ein Paritätsbit erweiterten— Datenstromes soll auf einer beliebigen Menge der vier Ausgangskanäle `DataOut0`–`DataOut3` erfolgen können. D.h. die Ausgabe kann beispielsweise nur auf Kanal 2 (unicast), parallel auf Kanal 1 und 3 (multicast) oder parallel auf alle Kanäle (broadcast) erfolgen.

Die durch ein Kommando eingestellte Auswahl bleibt bis zum nächsten entsprechenden Kommando gültig.

Die Kommandos sollen auf der gleichen Eingangsleitung wie die Daten selbst bereitgestellt werden. An der Eingangsleitung `DataControlIn` erscheint somit eine Bitfolge, die sowohl Kommandos als auch Datenblöcke enthält. Das erste Bit jeder Sequenz entscheidet, ob es sich bei den nachfolgenden Bits um einen Datenblock (0) oder um ein Kommando (1) handelt:

- *Übertragung eines Datenblockes*  
Jeder Datenblock hat die feste Länge von 4 Bits, es folgen am Eingang also vier Datenbits. Diese Datenbits sollen an die selektierten Ausgabekanäle weitergeleitet werden. Je nach aktueller Konfiguration wird das Paritätsbit angehängt.
- *Übertragung eines Kommandos*  
Die nachfolgenden Bits identifizieren das Kommando sowie ggf. dessen Parameter. Die Festlegung des Kommando- und Parameterformates, d.h. der Bitfolge, die ein Kommando identifiziert, kann frei gewählt werden.

Nach Abarbeitung dieser Schritte entscheidet das nächste Bit wiederum, ob ein Datenblock oder ein Kommando folgt.

Zur Synchronisation des Bausteines mit der Außenwelt werden zwei Steuerleitungen verwendet: `requestData` und `DataReady`. Der Baustein signalisiert seine Bereitschaft, ein Bit am Eingang entgegenzunehmen, indem er die Steuerleitung `requestData` auf 1 setzt. Daraufhin wird erwartet, daß von außen das nächste Bit an der Eingangsleitung bereitgestellt und die Steuerleitung `DataReady` auf 1 gesetzt wird. Sobald der Baustein eine 1 auf der Leitung `DataReady` erkennt, nimmt er das Datenbit ab und signalisiert dies nach außen, indem er die Leitung `requestData` wieder auf 0 setzt. Hierdurch wird der externen Logik die Gelegenheit gegeben, die Abnahme des Datenbits zu erkennen und das nachfolgende Datenbit vorzubereiten. Der Baustein bekommt somit seitens der externen Logik die Stabilität des Datenbits am Eingang in dem Zeitraum garantiert, in dem sowohl `requestData` als auch `DataReady` auf 1 ist.

Entwerfen Sie diesen Funktionsbaustein als mikroprogrammierte Zustandsmaschine. Sie können dabei beliebige aus der Vorlesung bekannte Bausteine wie beispielsweise Register, Multiplexer, usw. verwenden. Geben Sie in Ihrer Lösung ein Blockschaltbild des internen Aufbaus,

das Format der Mikrobefehle, das Mikroprogramm sowie eine kurze Beschreibung der Funktionsweise des von Ihnen entworfenen Bausteines an.

Ausgegeben: 14.01.2000

Abgabe: bis spätestens Freitag 21.01.2000 *vor* der Vorlesung