



# Algorithmenentwurf für massiv parallele Hardware

## Aufgaben zur Vorbereitung – Teil II

Bearbeitungszeit: bis Di. 01.05.2012

### Aufgabe 1 – Basiskommunikation

Erstellen Sie ein einfaches RIVYERA-Projekt, welches folgende Funktionen erfüllt.

Jeder FPGA merkt sich das zuletzt vom Host an ihn gesendete Datum. Eine Leseanfrage vom Host inkrementiert dieses Datum um eins und schickt es dann zurück. Eine weitere Anfrage inkrementiert es nochmals usw. Vereinfacht dürfen Sie annehmen, dass der Absender der Leseanfragen immer der Host ist, und dass immer nur genau ein Datum angefordert wird.

Testen Sie Ihr Design indem Sie vom Host aus beliebige Daten verschicken und wieder anfordern. Fordern Sie auch mehrfach Daten vom selben FPGA an.

**Hinweis:** Benutzen Sie für die Aufgaben des Hosts das Tool `se_mon`. Nach dem Start des Tools können Sie mit `help` eine kleine Hilfe aufrufen. Vergessen Sie nicht mit `allocMachine` die RIVYERA für sich zu reservieren. Vergessen Sie aber vor allem nicht sie nach Ihrem Test auch wieder mit `freeMachine` wieder für andere Nutzer freizugeben!

### Aufgabe 2 – FIFO

Implementieren Sie für einen RIVYERA-FPGA eine FIFO beliebiger Größe im Block-RAM. Benutzen Sie dazu den *Xilinx CoreGenerator*. Ihr FPGA-Design soll nun diese FIFO nutzen um Daten, die von einer beliebigen Quelle gesendet werden, zwischen zu speichern. Werden mehr Daten gesendet als in der FIFO Platz haben, werden die letzten Daten ignoriert.

Wird eine Leseanfrage gestellt, soll die angeforderte Menge an Daten aus der FIFO an die Quelle der Leseanfrage zurückgeschickt werden. Werden mehr Daten angefordert als in der FIFO verfügbar sind, so wird der Rest mit Nullwörtern aufgefüllt.

Testen Sie Ihr Design indem Sie ein Host-Programm schreiben, welches Arrays mit Zufallszahlen generiert, diese an die verfügbaren FPGAs verteilt und anschließend wieder von den FPGAs zurückfordert. Überprüfen Sie die empfangenen Daten, ob sie mit den versendeten übereinstimmen.