

Übung zur Vorlesung Architekturen und Entwurf von Rechnersystemen

Prof. Dr-Ing. A. Koch
Jaco Hofmann, MSc.



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Wintersemester 16/17
Übungsblatt 6

In den letzten Übungen wurden vor allem streambasierte Architekturen betrachtet. Dem gegenüber werden in System on Chips (SoCs) häufig Architekturen verwendet die auf einen gemeinsamen Speicher zugreifen und die über Busse konfiguriert werden können. Diese Übung widmet sich diesen Bussystemen durch die Entwicklung des ESA-Bus. Der ESA-Bus ist ein sehr einfach gehaltener Bus der vergleichbar ist mit AXI4-Lite. Lese- und Schreibrichtungen sind getrennt. Die Busbreite ist 8 bit und die Adressbreite sind 16 bit. Der Bus ist Byte-adressiert. Der Schreibkanal hat keine Rückrichtung. Außerdem fehlen, zur Vereinfachung, viele Signale, z.B. ein Tag, die ein Produktiv verwendeter Bus hat. Des weiteren sind keine Burst-Transfers erlaubt.

Signal	Beschreibung
raready	Slave kann Leseanfrage annehmen.
ravalid	Master sendet gültige Daten.
raaddr	Leseadresse
rrready	Master kann Antwort annehmen.
rrvalid	Slave möchte Antwort senden.
rrdata	Gelesene Daten.

Tabelle 1: Signale der Leserichtung

Signal	Beschreibung
wready	Slave kann Schreibenanfrage annehmen.
wvalid	Master sendet gültige Daten.
waddr	Schreibadresse
wdata	Schreibdaten

Tabelle 2: Signale der Schreibrichtung

Aufgabe 6.1 Master und Slave

Erstellen Sie einen passenden Master und Slave. Achten Sie darauf, dass Master und Slave möglichst leicht wieder zu verwenden ist. Ein Slave kann z.B. ein Speicher sein, aber auch die Konfigurationsregister eines Bildfiltermoduls.

Überlegen Sie sich geeignete Interfaces. Bedenken Sie, dass Bluespec bei Action/ActionValue Methoden automatisch Handshake-Signale generiert.

Aufgabe 6.2 Master und Slave verbinden

Erstellen Sie ein Modul das einen Master und einen Slave verbinden kann. Das Interface ist dabei Empty und Master und Slave werden als Parameter an das Modul übergeben. Sie können die Connectable-Typklasse aus der AzureIP Bibliothek verwenden.

Übung zur Vorlesung Architekturen und Entwurf von Rechnersystemen

Aufgabe 6.3 Speicherslave

Erstellen Sie einen Speicher mit dem ESA-Bus Interface. Der Speicher soll 4096 Einträge vom Typ `Bit#(8)` haben. Benutzen Sie als Grundlage des Speichers das `RegFile` aus der AzureIP Bibliothek. Details zur Verwendungen finden Sie im Bluespec Reference Guide.

Testen Sie die Kommunikation über den Bus mit `BlueCheck`. Vergleichen Sie dazu ein lokales `RegFile` mit dem über ESA-Bus angebundenen Speicher.

Aufgabe 6.4 Mehrere Master und Slaves (Schwierig)

Erstellen Sie ein Modul, dass die Kommunikation zwischen mehreren Slaves und Master regelt. Überlegen Sie sich ein geeignetes Arbitrationsverfahren (z.B. Round-Robin) um zu Entscheiden welcher Master aktiv sein darf. Jeder Slave bekommt einen 12 bit Adressraum. Die restlichen Bit der Adresse werden verwendet um den anzusprechenden Slave zu identifizieren.

Es darf immer nur ein Lesezugriff gleichzeitig aktiv sein, alle anderen Zugriffe werden pausiert. Die Zahl der Master und Slaves soll nicht vorgegeben sein. Übergeben Sie einen Vektor der Master und Slaves an das Modul.