Hochschule Osnabrück

Fakultät Ingenieurwissenschaften und Informatik

Labor für Digital- und Mikroprozessortechnik

Prof. Dr.-Ing. B. Lang

Praktikum Digitale Komponenten

Uebung 2: Endliche Automaten, digitale Verarbeitungssysteme, Testbenches

Aufgabenstellung

- 1. Beschreiben Sie die in der Vorlesung vorgestellte Komponente "Serieller_Sender" in VHDL
- 2. Erstellen Sie eine Komponente "Datengenerator", welche eine Zeichenkette als Datenstrom erzeugt
- 3. Verbinden Sie "Datengenerator" und "Serieller_Sender" zu einem "Sender_System"
- 4. Erzeugen Sie einen Bitstream und programmieren Sie damit ein FPGA

Aufgabe 1: "Serieller_Sender" erstellen und verifizieren

Erstellen Sie den in der Vorlesung vorgestellten seriellen Sender als VHDL-Komponente ("Serieller_Sender.vhd"). Beschreiben Sie das Rechenwerk und das Steuerwerk in jeweils einem eigenen Block (siehe Abbildung 1 und Abbildung 2). Vervollständigen Sie alle mit "TODO" markierten Stellen im Quellcode.

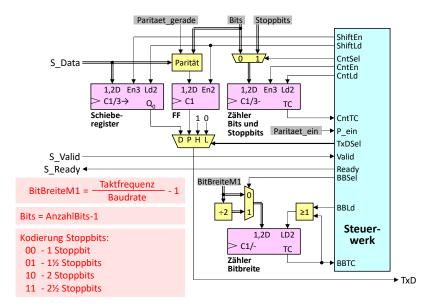


Abbildung 1: Struktur der Komponente "Serieller Sender"

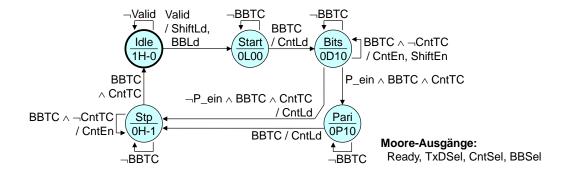


Abbildung 2: Steuerwerk der Komponente "Serieller Sender"

Simulieren Sie die Komponente mit der vorgegebenen Testbench "Serieller_Sender_tb.vhd". Nutzen Sie zur ihrer Ausführung das Script "test Serieller Sender.do" (siehe Abbildung 5).

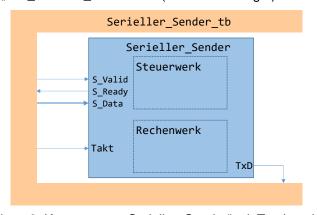


Abbildung 3: Komponente "Serieller_Sender" mit Testbench

Aufgabe 2: "Datengenerator" erstellen und verifizieren

Vervollständigen Sie die Komponente "Datengenerator" (Datei "Datengenerator.vhd"). Der auszugebende Text wird als generischer Parameter vom Typ "string" übergeben. Ein "string" ist ein Array von "character"-Elementen. Strings haben <u>positive</u> Indizes, die Zählung beginnt also bei 1! Die Umwandlung eines character c zu einem std_ulogic_vector v kann erfolgen durch:

```
v <= std_ulogic_vector(to_unsigned(character'Pos(c, v'Length));</pre>
```

Der Text wird zyklisch über eine Datenfluss-Schnittstelle ausgegeben. Simulieren Sie die Komponente mit der vorgegebenen Testbench "Datengenerator_tb.vhd". Nutzen Sie zur ihrer Ausführung das Script "test_Datengenerator.do" (siehe Abbildung 4).

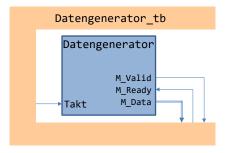


Abbildung 4: Komponente "Datengenerator" mit Testbench

Aufgabe 3: "Sender_System" erstellen und verifizieren

Verbinden Sie den Datengenerator mit dem Seriellen Sender zu einem Gesamtsystem (Datei "Sender_System.vhd"). Simulieren Sie die Komponente mit der vorgegebenen Testbench "Sender_System_tb.vhd". Nutzen Sie zur ihrer Ausführung das Script "test_Sender_System.do" (siehe Abbildung 5).

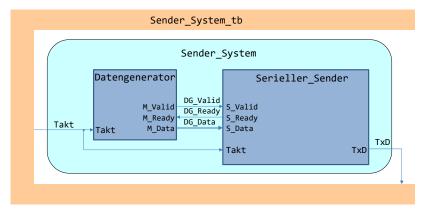


Abbildung 5: Struktur der Komponente "Sender System" mit Testbench

Aufgabe 4: Bitstream erzeugen und das FPGA programmieren

Starten Sie Vivado und erzeugen Sie ein neues Projekt für das Board Cmod-A7-35T (Falls das Board nicht zur Auswahl steht, können Sie auch ein Projekt für das "Part" xc7a35tcpg236-1 erstellen).

Fügen Sie folgende "Source Files" zum Projekt hinzu:

- Sender_System.vhd
- Datengenerator.vhd
- Serieller Sender.vhd

Fügen Sie folgende Constraints-Datei dem Projekt hinzu (diese enthält die Zuordnung der Ports zu den Pins des FPGA-Bausteins):

Sender_System.xdc

Erzeugen Sie nun durch Klicken auf 🏴 Generate Bitstream die Bitstream-Datei.

Verbinden Sie das FPGA-Board mit einem Micro-USB-Kabel mit dem PC. Verbinden Sie außerdem den USB-UART-Adapter einerseits mit einem Mini-USB-Kabel mit dem PC und andererseits mit Jumper-Kabeln mit dem FPGA-Board (siehe Abbildung 6).

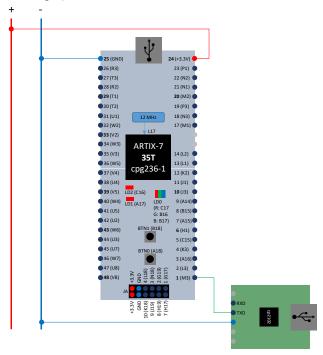


Abbildung 6: Verbindung von FPGA-Board und USB-UART-Adapter

Für den USB-UART-Adapter wird von Windows auf dem PC ein virtueller COM-Port eingerichtet, dessen Namen man im Windows-Geräte-Manager nachschauen kann. Auf dem Board befindet sich der Chip CP2102 von Silicon Labs (siehe Abbildung 7).

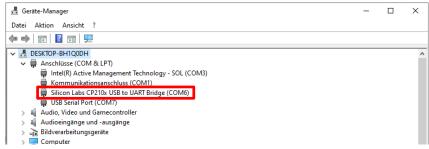


Abbildung 7: Virtueller COM-Port im Windows-Gerätemanager

Auf dem PC muss ein Terminal-Programm gestartet werden. Dazu können Sie sehr gut das kostenlose Programm HTerm verwenden (http://www.der-hammer.info/pages/terminal.html). Im Terminal-Programm müssen die Übertragungsparameter eingestellt werden. Diese sind der Datei Sender_System.vhd zu entnehmen. Weiterhin kann man einstellen, bei welchem empfangenen Zeichen ein Zeilenumbruch im Fenster erzeugt werden soll (siehe Abbildung 8)

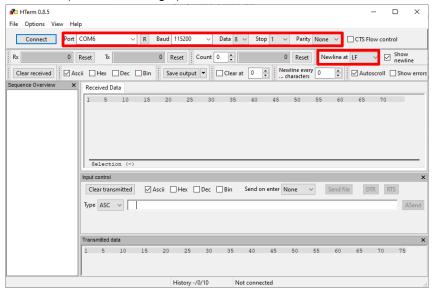


Abbildung 8: Konfiguration des Terminal-Programms HTerm

Nun kann das FPGA mit dem Bitstream konfiguriert werden. Dazu wird in Vivado der Hardware Manager geöffnet durch Klicken auf PROGRAM AND DEBUG→Open Hardware Manager→Open Target→Auto Connect. Die Programmierung erfolgt durch Klicken des Kommandos Program Device→xc7a35t_0. Im sich öffnenden Fenster wählt man den Button "Program" (siehe Abbildung 9). Die in der TCL-Konsole erscheinende Warnung "WARNING: [Labtools 27-3361] The debug hub core was not detected." Können Sie ignorieren.

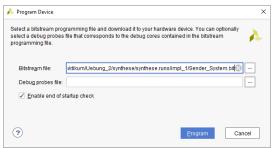


Abbildung 9: Programmierung des FPGAs

Im Terminal-Programm muss noch die Verbindung mit dem Button "Connect" hergestellt werden (siehe Abbildung 10), dann erscheinen die Ausgaben des FPGAs im Feld "Received Data".

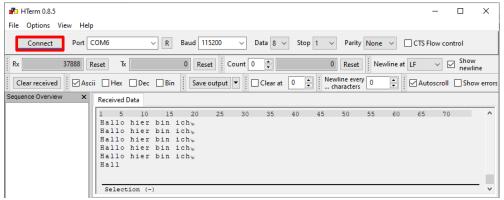


Abbildung 10: Ausgabe im Terminal-Programm