

Laborübung 2

Zu bearbeiten in der Woche vom Mittwoch, 4. Mai 2011, 16:00

Hinweis: Die Übungen sind Präsenzübungen, die vor Ort im Labor U09 in Gruppen von bis zu drei Studierenden bearbeitet werden können. Die Teilnahme an den Laborübungen ist freiwillig.

2.1 Lauflicht

Die SystemC-Library und der C++-Compiler g++ sind auf dem Linux-Rechner `ciuser08` verfügbar. Sie können sich mit dem Windows-Account für das CI-Lab auf dem Linux-Rechner von den CI-Lab-Rechnern aus mit Hilfe des X11-Servers anmelden (Anleitung umseitig).

1. Erstellen Sie eine SystemC-Version des in der vorigen Laborübung als Test verwendeten Lauflichts. Übersetzen Sie den SystemC-Code und simulieren Sie das System.
2. Erzeugen Sie nun mit den tracing-Funktionen von SystemC eine `vcd`-Datei und überprüfen Sie die Eingaben und zugehörigen Ausgaben in GTKwave.
3. Erstellen Sie eine einfache Testbench, die die korrekte Abfolge des Aufleuchtens der LEDs überprüft.

Hinweise:

- Verwenden Sie Datentypen, die auf `bool` basieren – nicht auf `sc_logic`.
- Implementieren Sie das *top-level*-Modul in einem eigenen `SC_MODULE`, verbinden Sie für die Simulation eine `sc_clock` in `sc_main`.
- Verwenden Sie zur Übersetzung den g++-Compiler. Die Optionen für Include- und Library-Pfade lauten:

```
g++ -I/opt/systemc/include -L/opt/systemc/lib-linux ....
```

2.2 Synthese

In diesem Schritt soll nun das erstellte System synthetisiert werden.

Das SystemC-Synthesewerkzeug ist ebenfalls auf dem Linux-Rechner `ciuser08` verfügbar. In den Accounts sind die entsprechenden Umgebungsvariablen bereits gesetzt. Sie können das (Eclipse-basierte) Synthesewerkzeug durch Eingabe von `autoesl` im Terminal aufrufen, das sich öffnende Fenster ist in Abb. 1 dargestellt.

Legen Sie ein neues Projekt an und importieren Sie die C++-Quelldateien Ihres SystemC-Lauflichts. Für die Synthese legen Sie eine neue *Solution* an und starten die Synthese. Beachten Sie das Console-Fenster und versuchen Sie, evtl. auftretende Fehler zu beheben. (Die im Error Log angezeigten Fehlermeldungen sind oft unvollständig und verwirren eher).

Das Ergebnis der Synthese ist im Projektverzeichnis unter `syn/` zu finden. Dort existiert ein Unterverzeichnis `vhdl/` mit dem erzeugten VHDL-Code.

Hinweis: Sie können Dateien zwischen den Windows-PCs und dem Linux-System mit dem Windows-Programm `winscp` kopieren.

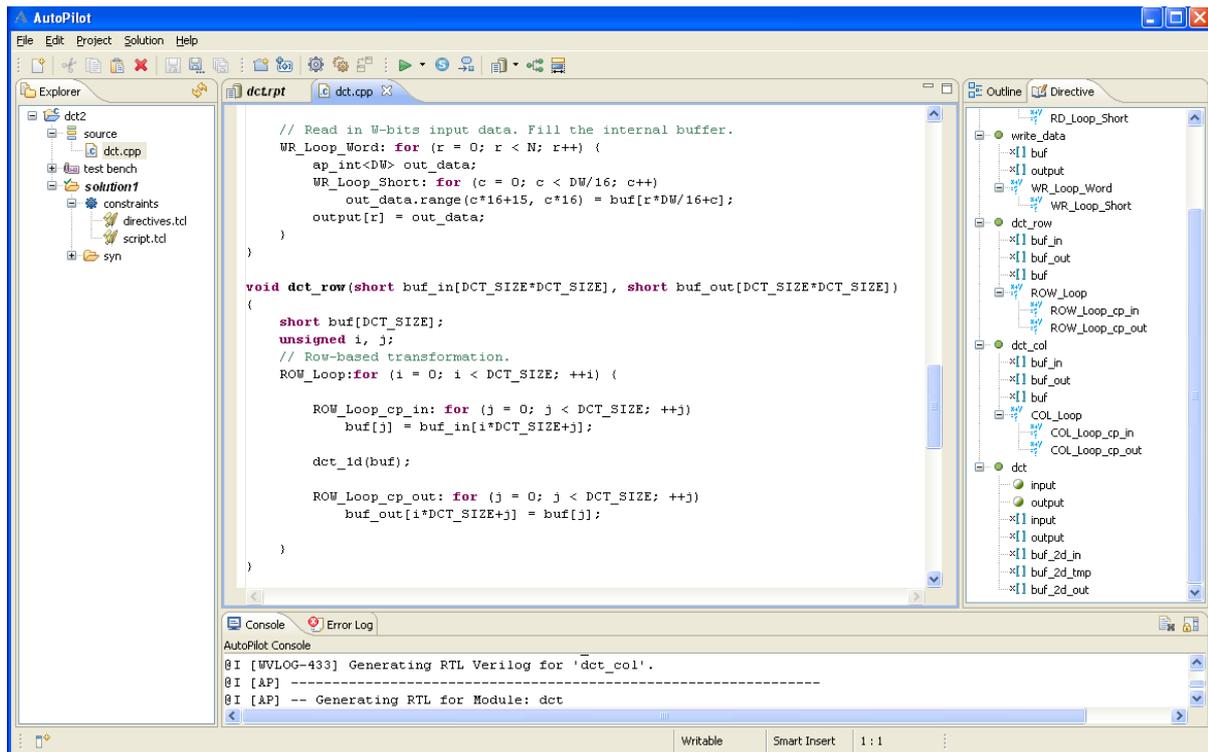


Abbildung 1: Screenshot AutoESL

2.3 Implementierung

Wenn die Synthese erfolgreich beendet wurde, können die erzeugten VHDL-Dateien auf den Windows-PC übertragen werden. Importieren Sie die Dateien (inklusive der automatisch erzeugten Datei `AESLcomponents.vhd`) in ein Xilinx ISE-Projekt. Hierzu können Sie z.B. das Projekt aus der vorigen Übung kopieren und anpassen.

Für die Synthese benötigen Sie noch die UCF-Datei, die die Zuordnung von Signalnamen zu FPGA-Pins angibt. Passen Sie hier die Datei aus der vorigen Übung entsprechend an.

Erzeugen Sie nun ein bitfile für das Virtex 5-Board und testen Sie das Lauflicht auf dem XUPv5-Board!

Anmeldung an ciuser08 mit X11-Forwarding

Zunächst muss auf Windows XP der X11-Server unter Start - Programme - Xming - Xming gestartet werden. Dann kann mit `putty` eine Verbindung zu `ciuser08` konfiguriert werden. In `putty` muss in der Konfiguration unter Connection - SSH - X11 die Option `Enable X11 Forwarding` aktiviert sein. Nach dem Verbindungsaufbau mit `putty` zu `ciuser08` können dann auch X11-Anwendungen wie `xterm`, `gtkwave` oder eben `autoesl` gestartet werden.