

1 Tool- und Hardwareübersicht

Für die Entwicklung der MIPS CPU wird in den Übungen das *Zedboard* (<http://www.zedboard.org>) eingesetzt, das einen *Zynq7020* (FPGA und Dual ARM A9 Cores) der Firma Xilinx enthält. Als Toolumgebungen kommt Vivado von Xilinx, sowie (optional) der Modelsim (Questasim) Simulator von Mentor Graphics zum Einsatz.

2 Rechnerzugang

- Wenn ihr die Tools mit GUI bedienen wollt kann z.B. das Xpra-Webinterface des CIPs genutzt werden:

```
https://remote.cip.cs.fau.de/
```

- Zugang zu den Laborrechnern multipro-r[1-3]p[1-3] erhaltet ihr über die cs3-login:

```
ssh (-X) -J ab12cdef@cs3-login.informatik.uni-erlangen.de ab12cdef@multipro-r2p1  
(mit -X für X Forwarding wenn ihr die GUI verwendet)
```

- Ihr könnt dabei euren IdM-Account verwenden mit dem Passwort welches im IdM für Linux-Systeme vergeben wurde.

3 Umgebungen setzen

- Damit alle Skripte richtig funktionieren verwendet am Besten die Bash:

– Bash starten: `bash`

- Vor dem Start der Tools Module einbinden.

- Verfügbare Module anzeigen:

```
module avail
```

- Profil für Vivado setzen:

```
module load Core/vivado/vivado2022.1
```

– Vivado starten: `vivado`

- (optional) Profil für Modelsim setzen:

```
module load Core/mentor/questasim/questasim2021.3_2
```

– (optional) `sim_signal_barrier.do` testen: `vsim -64 -do sim_signal_barrier.do`

4 Aufgaben

32-Bit Zähler (Programmzähler)

- Erstellen Sie ein VHDL-Modul, welches in jedem Taktzyklus einen 32-Bit Wert um 4 inkrementiert wird. Das Register des Zählers soll über ein Enable Bit *EN* verfügen. Nur bei anliegender '1' an diesem Port soll der Zustand des Registers inkrementiert werden. Über einen synchronen *Reset* soll der Zähler zurück auf 0 gesetzt werden können.
- Erweitern Sie Ihr Design um einen 32-Bit Dateneingang, um den Zähler auf einen beliebigen Wert zu setzen. Durch Setzen eines weiteren Steuerbits *LD* (Load) auf '1' soll der Wert, welcher am Dateneingang anliegt, im Register des Zählers gespeichert werden. Falls '0' an *LD* anliegt, soll wie oben beschrieben gezählt werden. Dies wird später für Sprünge benötigt.
- Achten sie dabei auf sinnvolle Abhängigkeiten zwischen *EN*, *LD* und *Reset*
- Schreiben Sie eine Testbench für Ihren Zähler und testen Sie die Komponente.
- Starten Sie Vivado wie oben beschrieben, laden sie alle benötigten VHDL Dateien und synthetisieren bzw. implementieren Sie Ihr Design. Analysieren Sie die generierten RTL- und Technologie Schematics, sowie die entstandene Ressourcenauslastung. Für welche Maximalfrequenz ist Ihr Design lauffähig?
- **optionaler Zusatz:** Schreiben Sie eine Toplevel-Datei, welche den Zähler instantiiert und die obersten 8 Bit des Zählerausgangs mit den LEDs des Zedboards verbindet. Die Clock sollte vom Quarz des Zedboards genutzt werden. Dateneingang und *LD* können auf konstante Werte gesetzt werden. *EN* und *Reset* können auf die Switches gelegt werden. Schreiben Sie ein XDC-Constraint-File und testen Sie Ihr Design auf dem Zedboard.