

4. Übungsblatt

Lösen Sie die folgenden Aufgaben:

- i) Führen Sie evtl. besprochene Änderungen an Ihrem Registerfile aus und optimieren Sie gegebenenfalls das Handling von Register 0. Verbessern Sie unbedingt die Testbench, um auch den Spezialfall des Registers 0 und den Schreib- bzw. Lesezugriff auf alle anderen Register zu testen (falls übersehen).
- ii) Unser FPGA enthält schon Blöcke mit dem Takte verändert werden können. Damit können aus dem 100Mhz-Takt an Pin E3 neue Takte erzeugt werden. Instanzieren Sie dazu einen MMCM-Block¹ (Mixed Mode Clock Manager auf Seite 451). Das können Sie in Vivado auf zwei Wegen erreichen. Einmal indem Sie den verwendeten Quellcode verwenden oder indem Sie im Vivado-IP Catalog den Clock Wizard verwenden. Wenn Sie den Quellcode einsetzen wollen, müssen Sie den Parameter `CLKFBOUT_MULT_F` anpassen. Erzeugen Sie einen 40Mhz Takt und speisen Sie diesen in Ihren Zähler von letzter Woche ein. Überprüfen Sie mit einer Messung (Oszilloskop), dass wirklich ein 40Mhz Takt an Ihren Zähler angelegt wird.
- iii) Implementieren Sie die ALU unserer CPU. Die ALU arbeitet mit defaultmäßig mit 32 Bit-Operanden und ist ein *kombinatorischer* Schaltkreis. Achten Sie darauf keine Latches zu erzeugen²! Erweitern Sie Ihre Implementierung mit einem generischen Parameter, sodass man die ALU auch mit 64 Bit-Operanden verwenden kann.

Unsere ALU kann addieren, subtrahieren (mit und ohne Vorzeichen), vorzeichenlose Werte nach links verschieben, vorzeichenbehaftete und nicht vorzeichenbehaftete Werte vergleichen und nach rechts verschieben. Weiterhin muss die ALU die XOR, OR und AND Operation implementieren. In allen anderen Fällen liefert die ALU den Wert 1.

Hinweis: Denken Sie über `switch/case`-Konstrukte und `numeric_std` von VHDL nach.

- iv) Entwerfen Sie eine Testbench für die ALU und automatisieren Sie den Buildprozess mit `make` oder einem geeigneten Skript, wenn Sie `ghdl` verwenden (Die Vivado-Nutzer müssen sich weiter mit der Maus behelfen). Einige Hörer haben bisher mühsam alle Kommandos immer wieder eingetippt und die Resultate in einer Waveform angesehen. Stoppen Sie diese Zeitverschwendung! Verwenden Sie auf jeden Fall auch `report` und `assert` für Ihre Testbench!

Besprechung und Abnahme am 26. November 2025

¹Die URL funktioniert mit Copy&Paste auf jeden Fall: https://www.amd.com/content/dam/xilinx/support/documents/sw_manuals/xilinx2019_1/ug953-vivado-7series-libraries.pdf

²Verwenden Sie Vivado um eine echte Synthese durchzuführen. Vivado ist auf den Rechnern im Labor installiert [https://doku.cs.hs-rm.de/doku.php?id=vivado_design_suite&s\[\]=vivado](https://doku.cs.hs-rm.de/doku.php?id=vivado_design_suite&s[]=vivado).