Hochschule RheinMain
Prof. Dr. Steffen Reith

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) 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 erzeugen[1]! 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.

iii) Entwerfen Sie eine Testbench für die ALU und automatisieren Sie den Buildprozess mit `make` oder einem geeigneten Skript. 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 20. November 2024

[1] Verwenden 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`.