

3. Übungsblatt

Lösen Sie die folgenden Aufgaben:

- i) Führen Sie evtl. besprochene Änderungen an Ihrem RAM aus. Verbessern Sie unbedingt die Testbench, um *alle* Speicherzellen zu testen. Verwenden Sie dazu `UNIFORM` der Library `IEEE.MATH_REAL` um Pseudozufallszahlen zu erzeugen. Stellen Sie insbesondere sicher, dass Ihre RAM das BlockRAM Ihres FPGAs verwendet.
- ii) Fixen Sie den Code des beigelegten Counters und fügen Sie einen `generic`-Parameter für die Breite ein. Implementieren Sie diesen Zähler auf Ihrem FPGA-Board, legen Sie vier Bits des Zählerwerts an vier LEDs. Führen Sie eine Messung an den LEDs mit Hilfe eines Oszilloskops durch. Decken sich die gemessenen Zeiten mit den erwarteten Zeiten? Wenn nein, dann finden Sie Gründe. Machen Sie Screenshots von der Messung für die Abnahme.
- iii) Implementieren Sie die Registerbank unserer RISC-V Implementierung. Unsere CPU hat 32 Register mit einer Breite von 32 Bit. Beachten Sie, dass wir gleichzeitig aus zwei Registern lesen und in ein Register schreiben können müssen (siehe dem Architekturbild von Ripes¹ für die Single Cycle Version eines RISC-V Prozessors). Überlegen Sie, warum dies notwendig ist. Achtung: Register 0 liefert beim Lesen *immer* den Wert 0 und geschriebene Daten werden verworfen. Versuchen Sie so wenig wie möglich FlipFlops für Ihr Design zu verbrauchen.
- iv) Entwerfen Sie eine Testbench für die Registerbank und automatisieren Sie den Buildprozess mit `make` oder einem geeigneten Skript. Ihre Testbench muss insbesondere die Funktion von Register 0 testen. Stellen Sie sicher, dass Ihre Testbench automatisiert laufen kann, d.h. die Korrektheit wird in der Simulation ermittelt und eine manuelle Überprüfung der Waveform ist nicht notwendig!

Besprechung und Abnahme am 19. November 2025

¹<https://github.com/mortbopet/Ripes>