

5. Übungsblatt

Lösen Sie die folgenden Aufgaben:

- i) Aktualisieren Sie Ihre Projektkopie mit `git`. Überarbeiten Sie gegebenenfalls Ihren Registerfile (Verwenden Sie z.B. sprechende Namen für Konstanten).

Das Register `r0` liefert laut Spezifikation immer den Wert 0 und kann nicht beschrieben werden. Fixen Sie Ihre Implementierung, wenn Sie dies übersehen haben. Arbeiten Sie die besprochenen Änderungen für `r0` ein.

Hinweis: Erzeugen Sie kein kompliziertes Enable-Signal für das Schreiben in das Registerfile und die Behandlung von Register 0, sondern geben Sie den Registerinhalt bzw. den Wert 0 aus, wenn das Register 0 adressiert wurde. Achten Sie auf den richtigen Indexbereich des verwendeten Arrays, um zusätzliche Subtrahierer zu vermeiden.

- ii) Implementieren Sie das Modul „Datapath“ der HaDes-CPU. Für die ALU können Sie die beschriebenen Komponenten aus der HaDes-Bibliothek verwenden oder gleich die Funktionalität von `numeric_std`.

Hinweis: Bei der Implementierung des Registerfiles haben Sie gesehen, wie die notwendigen Flip-Flops beschrieben werden, d.h. die vielen FlopFlops um die ALU herum können recht einfach beschrieben werden.

Hinweis: Es *kann* von Vorteil sein die Zwischenstände Ihrer Implementierung zu synthetisieren, um immer wieder mit Abbildung 4.9 der HaDes-Dokumentation zu vergleichen! In diesem Fall müssen Sie aber mit Vivado arbeiten (Synthese machen und den Schaltplan ansehen) .

Hinweis: Evtl. macht es Sinn dieses kompliziertere Modul mit mehreren Entities zu entwerfen.

Besprechung und Abnahme in der KW51 am 16. Dezember 2020.