

## 5. Übungsblatt

Lösen Sie die folgenden Aufgaben:

- i) Führen Sie evtl. besprochene Änderungen an Ihrer ALU aus. Verbessern Sie unbedingt die Testbench! Jeder Funktion Ihrer ALU (addieren, subtrahieren, etc.) sollte mit mindestens 20 (möglichst zufällig gewählten) Werten getestet werden.
- ii) Implementieren Sie den Programmzähler PC unserer CPU. Dabei macht es Sinn die drei Teilkomponenten des Programmzählers aus Abbildung 1 in einer Komponente zu implementieren.  
Hinweis: Denken Sie nochmal darüber nach, wie Sie Ihren Speicher adressieren (Wort/Byte).
- iii) Entwerfen Sie eine Testbench für den PC und automatisieren Sie den Buildprozess mit `make` oder einem geeigneten Skript. Verwenden Sie auf jeden Fall auch `report` und `assert` für Ihre Testbench, um die Tests automatisieren zu können!
- iv) Implementieren Sie eine Teilfunktionalität des Befehlsdekoders unserer CPU. Zum aktuellen Zeitpunkt müssen nur R-Type Befehle dekodiert werden. Diese haben immer die Form  $R_i = R_j \text{ op } R_k$ . Bestimmen Sie dazu die Indices der beteiligten Register aus der Instruktion. Es bietet sich an, dass Sie Ihre Implementation so aufbauen, dass wir diese später leicht durch andere Befehlsformate erweitern können (z.B. Sprünge).  
Hinweis: Spätestens nun sollten Sie sich auf die Suche nach geeigneter Literatur über RISC-V begeben (Buch!).

Besprechung und Abnahme am 1. Dezember 2023

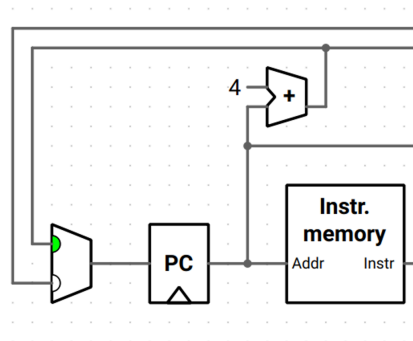


Abbildung 1: Ausschnitt Ripes - Single Cycle CPU