

## Laborprojekt – Versuch 4

In Ihren letzten Versuchen haben Sie alle notwendigen Komponenten für die Umsetzung einer Register-Befehl basierten Mikroarchitektur erstellt. In diesem Versuch fügen Sie diese zu einer ersten Umsetzung der RISC-V mit Pipelining zusammen. Auch wenn Teile der Pipeline für die Register-Befehle nicht notwendig sind, werden wir diese gleich zu Beginn komplett abbilden, um die daraus resultierende Struktur stückweise, angepasst an die Vorlesung, zu erweitern und wiederzuverwenden.

### Vorbereitung

#### Spezifikation RISC-V (R-Befehle)

Eignen Sie sich den Ablauf, Phasen und Ansteuerung für die Ausführung von Register-Befehlen basierend auf der Vorlesung und der Referenz (RV32I Base Integer Instruction Set, Version 2.1) an.

#### GHDL-Standards

Lesen Sie sich in die GHDL-Optionen ein, insbesondere wie Sie Standards der Sprache einstellen können. Ab sofort werden alle Komponenten und Testbenches nur noch mit VHDL 2008 verwendet.

#### Zusatzdateien

Im Rahmen dieses Versuchs stellen wir Ihnen die Dateien *ControlWordRegister.vhdl* und *instruction\_cache.vhdl* zur Verfügung. Während *ControlWordRegister.vhdl* ähnlich dem generischen Register aus Versuch 3 für das Steuerwort (ControlWord) ist, stellen wir Ihnen mit der Datei *instruction\_cache.vhdl* einen vereinfachten (read only) Cache für die Befehle zur Verfügung. Machen Sie sich mit den Register-Befehlen, und wie Sie diese codieren und im Instruction-Cache hinterlegen können, vertraut.

# Aufgaben

## Aufgabe 1: Erweitern des Registerfiles .....

Für den heutigen Versuch benötigen wir eine Möglichkeit, die Ergebnisse der Register-Befehle in einer Testbench zu prüfen. Erweitern Sie dazu die Datei *register\_file.vhdl*, um einen weiteren Ausgang *po\_registerOut* vom Typ *registerMemory*. Auf diesem soll durchgängig der komplette Inhalt des Register-Files (*s\_registers*) ausgegeben werden.

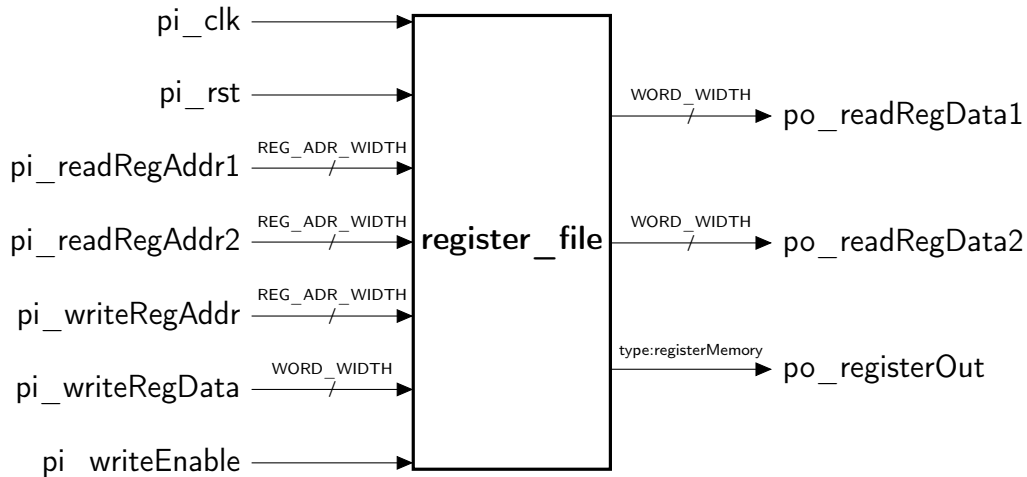


Abbildung 1: register\_file

Die Änderung sollte keinen Einfluss auf Ihre bereits erstellte Testbench des Register-Files haben.

- Ändern Sie die Entity *register\_file* in der Datei *register\_file.vhdl* und setzen Sie das beschriebene Verhalten um.

## Aufgabe 2: RISC-V für R-Befehle .....

Die Abbildung 2 zeigt die Umsetzung der RISC-V für Register-Befehle basierend auf der Vorlesung Rechnerarchitektur zur Umsetzung in VHDL.. Setzen Sie diese wie folgend um:

- Erstellen Sie den Unterordner *<Projektordner>/Komponenten/Cache* und Kopieren Sie die Datei *instruction\_cache.vhdl* dorthin.
- Kopieren Sie sich die Datei *ControlWordRegister.vhdl* in den Unterordner *<Projektordner>/Komponenten/Register*.
- Kopieren Sie sich die Datei *r\_only\_RISC\_V\_tb.vhdl* in den Unterordner *<Projektordner>/Testbench/RISCV*.

- Kopieren Sie sich die Datei *r\_only\_RISC\_V.vhdl* in den Unterordner *<Projektordner>/RISCV*.
- Erweitern Sie die Datei *r\_only\_RISC\_V.vhdl* gemäß den Vorgaben und orientieren Sie sich dabei an der vorgehenden Abbildung 2. Nutzen Sie für die Verknüpfungen immer separate und eindeutig benannte Signale. Alle Takteingänge werden mit *s\_clk* angesteuert. Ausgenommen ist der Takteingang des Registerfiles und des Instruktion-Caches. Diese werden mit der invertierten Clock (*not s\_clk*) angesteuert.
- Sie könnten die Abbildung auch reduzieren in dem Sie eine Befehls-Phase auslassen, dazu müssen Sie begründen warum Sie diese weglassen können, was in der Testbench geändert werden müsste und welche Auswirkungen das hat.
- Die Eingabe für den Port *pi\_instructionCache* erfolgt über das Signal *s\_instructions* und für *po\_registerOut* über das Signal *s\_registersOut*.
- Überprüfen Sie ihre Umsetzung durch das Ausführen der Testbench *r\_only\_RISC\_V\_tb* in der Datei *r\_only\_RISC\_V\_tb.vhdl* erfolgreich.
- Erstellen Sie ein Bash-Skript im Ordner *<Projektordner>/Skripts/RISCV*, mit dem Sie die Simulation der RISC-V für Register-Befehle inklusive der Analyse und Elaboration automatisch und erfolgreich ausführen können.

**Abnahme** Funktion des RISC-V für Register-Befehle, Programmierkonventionen, RISC-V Spezifikation der Register-Befehle.

### Aufgabe 3: Register-Befehle für Anfänger .....

Erstellen Sie mit der vorhandenen Umsetzung ein Programm, welches 5 mal das Register 1 zum Register 2 hinzuaddiert. Erweitern Sie dazu die Testbench (*r\_only\_RISC\_V\_tb2.vhdl*) so, dass die entsprechenden Befehle im Speicher vorgehalten, nach jedem Takt die aktuell berechnete Zahl ausgegeben und abschließend geprüft wird, ob die berechnete Zahl dem erwarteten Ergebnis entspricht. Erweitern Sie dazu das bereits erstellte Bash-Skript.

**Abnahme** Programmierkonventionen, RISC-V Spezifikation der Register-Befehle.

### Aufgabe 4: Abgabe.....

Laden Sie die erstellte Ordnerstruktur mit den dazugehörigen Dateien als Zip-Datei unter dem Namen

*Name\_Vorname\_Versuch4.zip* in Moodle hoch.

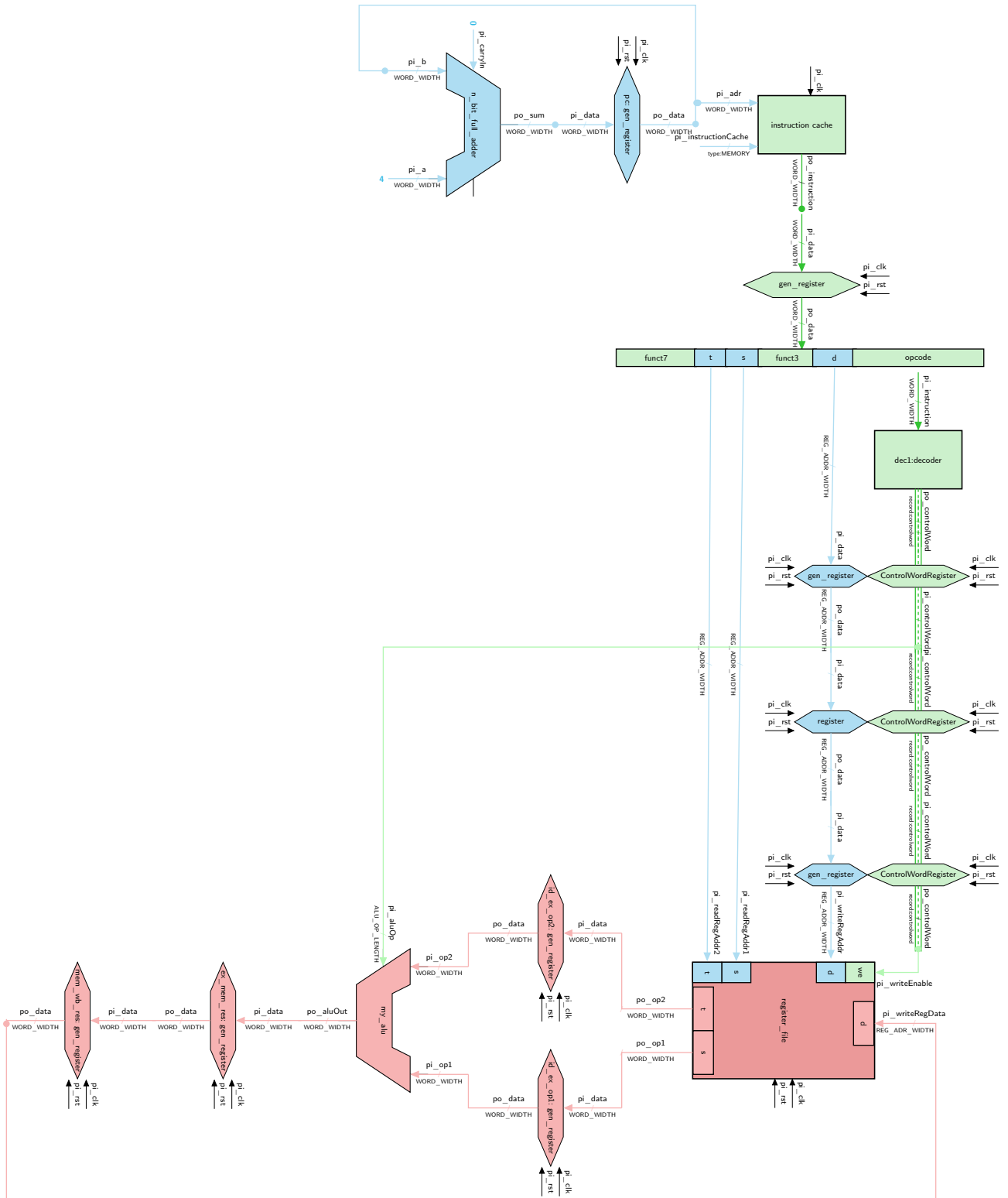


Abbildung 2: Mikroarchitektur der RISC-V beschränkt auf Register-Befehle