

## Laborprojekt – Versuch 8

Dieser Versuch beinhaltet die Umsetzung des Datenspeichers in der Memory-Phase als idealisierte Komponente. Idealisiert bedeutet, dass der Speicher mit allen nötigen Daten direkt eingebunden wird. In einem komplexeren System würde an dieser Stelle ein Cache mit Anbindung an die nächste Speicherstufe stehen, um die Zeiten und Kosten des Systems zu reduzieren. Sie werden dazu in diesem Versuch die Load- und Store-Befehle umsetzen, die das Speichern und Laden, also den Austausch von Daten zwischen Registerfile und Datenspeicher, ermöglichen.

### Ziele des 8. Versuchs

- Erweiterung der RISC-V-Umsetzung um Load- und Store-Befehle (L/S-Befehle)
- Verständnis für den Aufbau und die Funktion eines idealisierten Datenspeichers
- Integration des Datenspeichers in die Memory-Phase einer 5-Stufen-Pipeline
- Umsetzung des Datentransfers zwischen Registerfile und Speicher (inkl. Steuerung)
- Anwendung und Test der Implementierung mit Hilfe von GHDL und einer Testbench

### Vorbereitung

#### Leitfragen zur Vorbereitung

- Wie funktionieren Load- und Store-Befehle auf Ebene des Steuerwerks?
- Welche Rolle spielt das Immediate (I/S-Format) bei L/S-Befehlen?
- Wie beeinflussen die funct3-Bits das Verhalten des Speichers?
- Welche Signale müssen in der MEM- und WB-Phase gesetzt werden?

## Spezifikation RISC-V (L/S-Befehle)

Eignen Sie sich den Ablauf, die Phasen und die Ansteuerung für die Ausführung von Load- und Store-Befehlen basierend auf der Vorlesung und der Referenz (RV32I Base Integer Instruction Set, Version 2.0) an.

Eignen Sie sich ein Grundverständnis so an, dass Sie die Befehlskodierung und die Ansteuerung mithilfe der Referenzkarten für jede Phase wiedergeben und erklären können. Erarbeiten Sie sich ein Konzept zur Umsetzung der RISC-V-Befehle zum Speichern und Laden.

Sie sollten in der Lage sein, die Notwendigkeit von Caches in Bezug zum Pipelining und den Unterschied zwischen Caches und idealisierten Speichern erklären zu können und die verbundene Problematik aufzeigen zu können.

## GHDL-Standards

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

## Aufgaben

### Aufgabe 1: L/S-Befehle.....

Für die Umsetzung des Datenspeichers stellen wir Ihnen die Entity *data\_memory* zur Verfügung.

Diese verfügt über Ports, die zum Lesen und Schreiben der Daten auf dem Speicher sowie dem Steuereingang zur Befehls-abhängigen Selektion und Erweiterung der Datenwörter (Halbwörter oder Bytes) dienen.

Diese Erweiterung ist normalerweise vor- und nachgelagert, kann aber durch die Idealisierung hier kompakter verwendet werden.

- Erweitern Sie Ihren Decoder um die Load- und Store-Befehle. Achten Sie darauf, dass Sie die Signale *I\_IMM\_SEL* und *MEM\_WRITE* für die Store-Befehle auf 1 setzen und für die Lade-Befehle *I\_IMM\_SEL*, *MEM\_READ* und *REG\_WRITE* jeweils auf 1 und *WB\_SEL* auf 11 (den bisher ungenutzten vierten Eingang des WB-MUX). Vergessen Sie abschließend nicht, das Signal *MEM\_CTR* gemäß der L/S-Befehle zu setzen (funct3, die Konstanten wurden dazu erweitert).
- Testen Sie Ihren Decoder mit der Testbench *decoder\_tb.vhdl* erfolgreich.
- Erweitern Sie Ihr System um den Datenspeicher in der Memory-Phase.

- Passen Sie Ihr System so an, dass die Ports des Datenspeichers wie folgt verbunden werden:
  - *pi\_writedata* mit dem zweiten Operanden des Registerfiles in der MEM-Phase
  - *pi\_adr* mit dem Ergebnis der ALU in der MEM-Phase
  - den notwendigen Steuersignalen in der MEM-Phase
  - den Ausgang *po\_readdata* über ein Pipeline-Register in der WB-Phase mit dem vierten Eingang des WB-MUX
  - der Takteingang wird wie der Befehlsspeicher mit dem invertierten Takt angesteuert

Signal	Beschreibung
REG_WRITE	aktiviert Schreibzugriff auf das Registerfile
MEM_READ	aktiviert Lesezugriff auf den Datenspeicher
MEM_WRITE	aktiviert Schreibzugriff auf den Datenspeicher
MEM_CTR	steuert Byte/Halfword/Word-Auswahl
WB_SEL = 11	Daten aus MEM an den WB-MUX

Tabelle 1: Erweiterung des Steuerworts für L/S-Befehle

- Testen Sie abschließend Ihr System *riubs\_only\_RISC\_V* mit der Testbench *riubs\_only\_RISC\_V\_tb.vhdl* erfolgreich.

**Hinweis zur Fehlersuche:** Sollten Sie beim Testen fehlerhafte Werte im Registerfile oder Speicher beobachten, prüfen Sie:

- ob MEM\_WRITE / MEM\_READ korrekt gesetzt sind
- ob das Immediate korrekt erweitert wurde
- ob der WB\_SEL-Multiplexer korrekt gesteuert wird

## Aufgabe 2: Abgabe.....

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

*Name\_Vorname\_Versuch8.zip* in Moodle hoch.