Rechnerarchitektur
Sommersemester 24

ertis embedded real-time systems
ulm university institute

Laborprojekt – Versuch 3

In Ihrem letzten Versuch haben Sie die generischen Komponenten ALU, 2-1-Multiplexer und Pipeline-Register erstellt. In diesem Versuch erweitern wir die Komponenten um ein Registerfile für den RISC-V (ähnlich dem der MIPS aus GdTi), ein Befehls-Decoder (beschränkt auf die Registerbefehle) und eine Sign-Extension. Die Sign-Extension wird ausgelegt auf alle Befehlsarten und soll die möglichen Immediates zur Verwendung in der RISC-V-Architektur auf 32-Bit gemäß Spezifikation erweitern.

Ziele des 3. Versuchs

- Konzeptionierung, Umsetzung und Validierung einer RISC-V beschränkt auf Register-Befehle

Vorbereitung

Spezifikation RISC-V (R/I-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 2.0) an. Eignen Sie sich ein Grundverständnis so an, dass Sie die Befehlscodierung und die Ansteuerung mithilfe der Referenzkarten für jede Phase wiedergeben und erklären können. Erarbeiten Sie sich ein Konzept zur Umsetzung der RISC-V für Registerbefehle.

GHDL-Standards

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

Zusatzdateien

Im Rahmen diese Versuches stellen wir ihnen die Dateien *ControlWordRegister.vhdl* und *instruction_cache.vhdl* zur Verfügung. Während *ControlWordRegister.vhdl* ähnlich dem generischen Register aus Versuch 2 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 Befehlen und wie Sie Register-Befehle codieren und in dem Instruction-Cache hinterlegen können vertraut.

Aufgaben

Aufgabe 1: Erweitern des Registerfiles

Für den heutigen Versuch benötigen wir eine Möglichkeit die Ergebnisse der Registerbefehle in einer Testbench zu Prüfen. Dazu verändern/erweitern Sie die Datei *register_file.vhdl* so, das Sie einen weiteren Ausgang *po_registerOut* vom Typ *registerMemory* hinzufügen. Auf diesen wird durchgängig der komplette Inhalt des Register-Files (*s_registers*) ausgegeben.

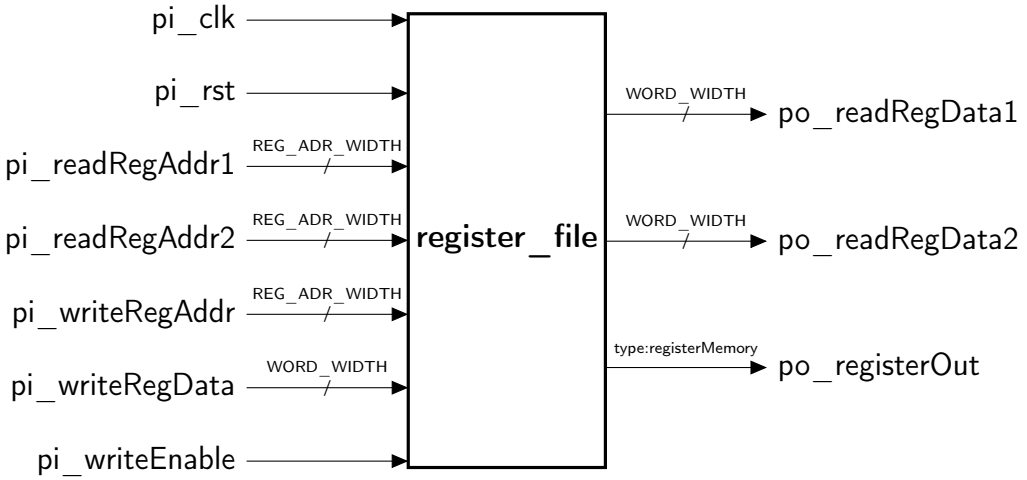


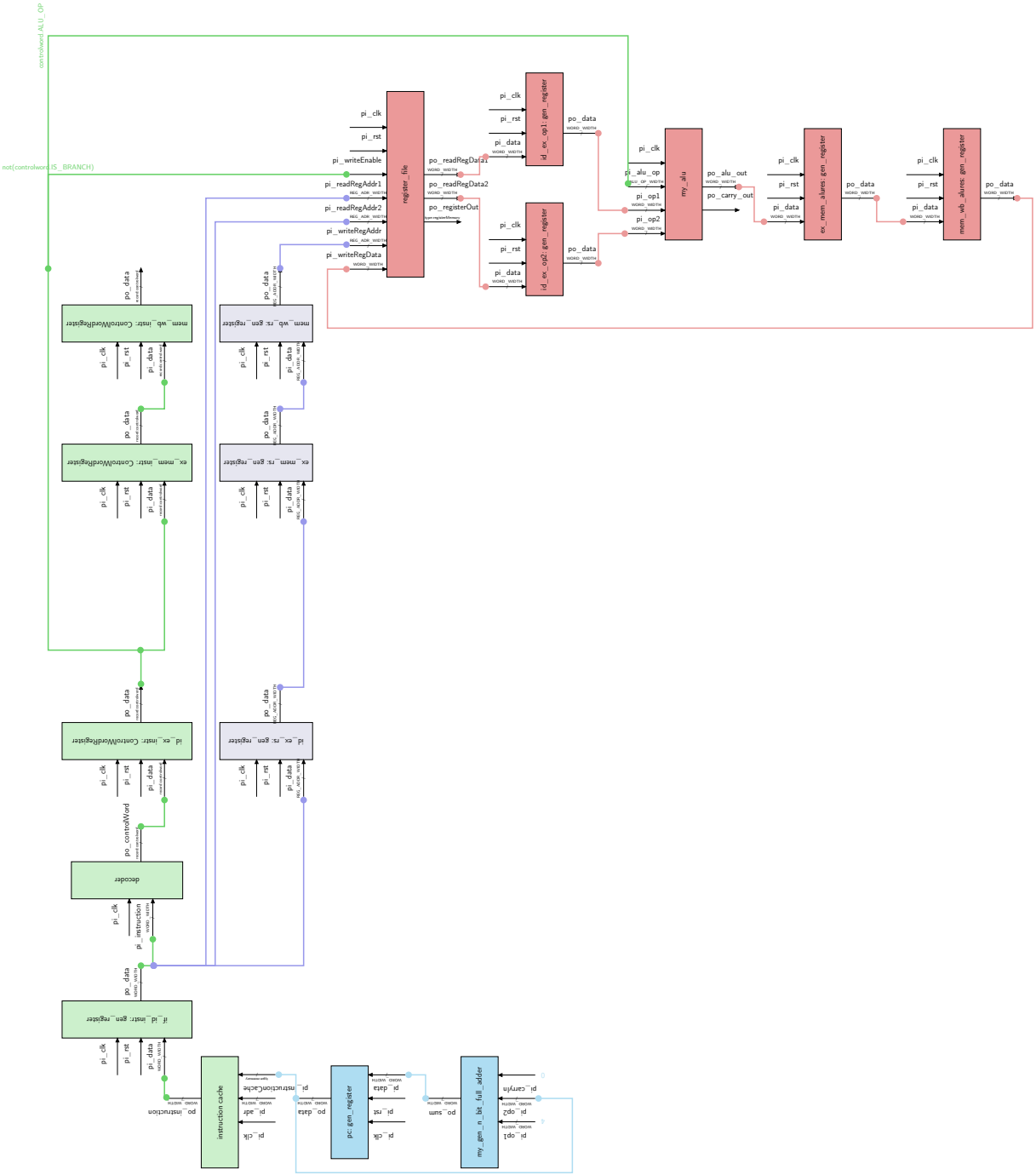
Abbildung 1: register_file

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. Initialisieren Sie zusätzlich das Registerfile mit dem Wert 9_{10} in Register 1 und dem Wert 8_{10} in Register 2.
- Testen Sie Ihre Änderung erfolgreich.

Aufgabe 2: RISC-V für R-Befehle

Die folgende Abbildung zeigt die Umsetzung der RISC-V für Registerbefehle und deren verwendeten Signale und Entities.



- 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*.

- Erweitern Sie die Datei *r_only_RISC_V_tb.vhdl* gemäß der Vorgegeben Abbildung. Orientieren Sie sich dabei an der vorgebenden Abbildung. Nutzen Sie für die Verknüpfungen immer separate und eindeutig benannte Signale. Alle Takteingäng der Register werden mit dem zur Verfügung gestellten Signal *s_clk2* und alle anderen mit *s_clk* angesteuert. Diese unterscheiden sich durch eine leichten Phasenunterschied.

- Sie können die Abbildung auch reduzieren in dem Sie eine Befehls-Phase auslassen, dazu müssen Sie begründen warum Sie diese weglassen können, in der Testbench ändern müssen und welche Auswirkungen das hat.

- Die Eingabe für den Port *pi_instructionCache* ist mit dem Signal *s_instructions* und für *po_registerOut* 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>/Testbenches/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 Registerbefehle.

Aufgabe 3: Registerbefehle für Anfänger

Erstellen Sie Ein Programm das mit der Vorhandenen Umsetzung die 5 mal das Register 1 zum Register 2 hinzuaddiert. Erstellen Sie dazu eine zusätzliche Testbench die die entsprechenden Befehle enthält, nach jedem Takt die aktuell berechnete Zahl ausgibt und abschließend prüft ob die berechnete Zahl dem erwarteten Ergebnis entspricht.

Abnahme Programmierkonventionen, RISC-V Spezifikation der Registerbefehle.

Aufgabe 4: Abgabe

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

Name_Vorname_Versuch2.zip in Moodle hoch.