

18.145 Rechnerarchitekturen und Mikrosystemtechnik

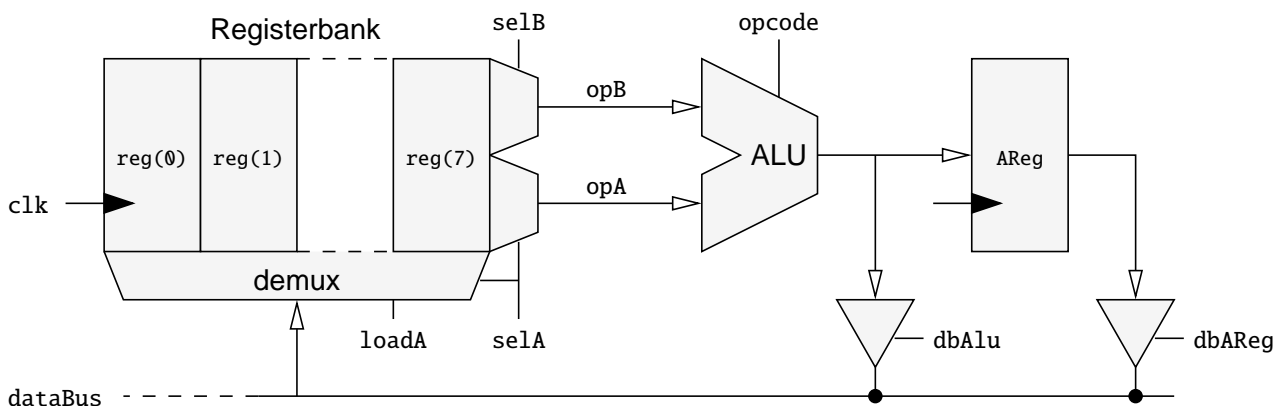


Übung: 2 VHDL Entwurf – Datenpfad

A. Mäder

Name	Matrikelnummer

Beim Entwurf von Mikroprozessoren wird in der Regel ein Kompromiss zwischen Zielvorgaben, Gestaltungsgrundsätzen und Randbedingungen angestrebt. Die gängigen Entwurfs- und Simulationswerkzeuge helfen bei der Entscheidung für oder gegen bestimmte Entwurfsalternativen. Das Ziel dieser Übung ist der Entwurf eines einfachen Mikroprozessorkerns am Beispiel eines Datenpfades im Operationswerk. Die Struktur auf Register-Transfer Ebene ist in folgender Skizze dargestellt.



Zum Verständnis dieser Aufgaben wird die Funktionsweise der einzelnen Einheiten kurz erläutert:

Registerbank bestehend aus 8 16-bit Registern. Ist `loadA` aktiv, dann wird das mit `selA` selektierte Register (`reg(selA)`) mit dem Wert des Datenbusses (`dataBus`) geladen.

ALU mit einigen einfachen Befehlen (siehe Tabelle), die als `opcode` spezifiziert werden. Die beiden Operanden `opA` und `opB` werden über Multiplexer (`selA`, bzw. `selB`) mit Werten aus der Registerbank versorgt. Für die arithmetischen Operationen wird eine 16-bit 2-Komplement („normale“ signed) Arithmetik zugrunde gelegt.

Funktionen der ALU:

- opB
- opB + 1
- opB - 1
- opA + opB
- opA + opB + 1
- opA - opB
- opB - opA
- opA and opB
- opA or opB
- opA xor opB
- opB nach links verschoben, '0' ergänzt
- opB nach rechts verschoben, '0' ergänzt
- opB nach rechts verschoben, MSB ergänzt
- opB nach rechts rotiert
- opB bitweise invertiert
- opB negiert

Ausgangsregister (16-bit), der ALU nachgeschaltet.

Bidirektionales Bussystem dataBus, das hier im Datenpfad zwei Quellen besitzt: den direkten Ausgang der ALU oder das dahinter liegende Register. Beide sind über Tristate-Treiber, die über entsprechende Kontrollsignale (dbAlu und dbAReg) selektiert werden, an den Bus angeschlossen.

Vorbereitung

Informationen zu der Sprache VHDL sind den Vorlesungsunterlagen zu entnehmen und finden sich auf der Web-Seite unter:

„**VHDL Kompakt**“ Einführung in die Konzepte und Nachschlagehilfe zur Syntax von VHDL.

„**VHDL Archiv**“ mit weiteren Informationen und Verweise, beispielsweise auf Online-Tutorials.

Aufgaben

Die Aufgabenteile 1 und 2 beinhalten einige Vorüberlegungen und sind schriftlich zu lösen – kurze Antwort auf diesem Zettel. Zum Übungstermin sollen Sie den VHDL-Code aus Aufgabe 3 selber simulieren und so die Syntax und die Funktionalität verifizieren. Den Code können Sie schon vorbereitend eingeben (mit einem beliebigen Texteditor) und auf Ihren Studentenaccounts speichern.

II-1 Wie würde auf der Architektur aus Abbildung 1 ein 3-Adress Befehl (Registeradressen) abgearbeitet werden?

Beispielsweise die Addition: `add(0, 1, 5)`

Funktion: `reg(0) + reg(1) → reg(5)`

Welche Verbesserungen in der Architektur wären demnach möglich?

II-2 Welche VHDL-Datentypen sind für die oben beschriebene Arithmetik (16-Bit, 2-Komplement) geeignet?

II-3 Codieren Sie die Schnittstellenbeschreibung (ENTITY) und das zugehörigen Verhalten (ARCHITECTURE) für den Datenpfad Prozessorkern.

Für die Simulation der Schaltung wird zum Übungstermin eine entsprechende Simulationsumgebung zur bereitgestellt. Um Ihr VHDL-Modell benutzen zu können, muss für die Instanziierung in der Testumgebung eine geeignete Komponentendeklaration vorhanden sein. Erstellen Sie deshalb auch eine Datei mit „passender“ Deklaration, die dann als Vorlage benutzt werden kann.