Hardware Design Lab 2017
Short Report

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Group 6: Lukas Boland, Sven Ströher, Ana Carolina Ferreira, Julian Käuser
Supervisor: M.Sc. Sarath Kundumattathil Mohanan, M.Sc.

Start: 08/01/2017 | Submission: 08/11/2017

Institute of Computer Engineering | Integrated Electronic Systems Lab
Prof. Dr.-Ing. Klaus Hofmann

1 Overview

According to the task given in the manual, a general purpose processor implementing the ARM Thumb-2 has been designed. All operations specified in this architecture, except `CBNZ` and `CBZ`, and operations on processor state change, are implemented. Both the `count32` and the `memcp46` benchmark applications can be executed.

The processor consists of the following larger submodules, as pointed out in Fig. 1:

- Instruction Decoder
- Register File
- Instruction Fetch Unit
- ALU (purely combinatorial)
- Flag Updater (purely combinatorial)
- Memory Interface

1.1 Pipeline Stages

There are two pipeline stages: one pipeline register is at the decoder output, the other at the output of the register file. In short, the stages could be called *fetch/decode* and *execute/writeback*.

1.2 Control

The whole processor is controlled by the decoder. This decision has been taken since we encountered that the most control decisions depend on complex instructions.

1.3 Memory Access

All memory access are abstracted from the rest of the CPU through a dedicated memory interface. This interface registers simple read/write request signals and executes the memory access. Addresses in the THUMB architecture are byte aligned, but the memory is halfword aligned. Therefore, the conversion of the requested address to the required address (-es) is performed by this module Additionally, sign extensions can be done on the data output.

1.4 ALU and Flags

All arithmetic operations are executed by a combinatorial ALU. It also outputs the values for the required flags (C, V, Z, N). The flags are only updated by a flag updater if the instruction requires that.

1.5 Instruction Decoder

The Instruction Decoder outputs all control signals one cycle after the instruction has been assigned. Furthermore, it has a state machine which manages the control signals in the following cycles.If necessary (at memory or multi-instructions), it may stall the instruction fetch.

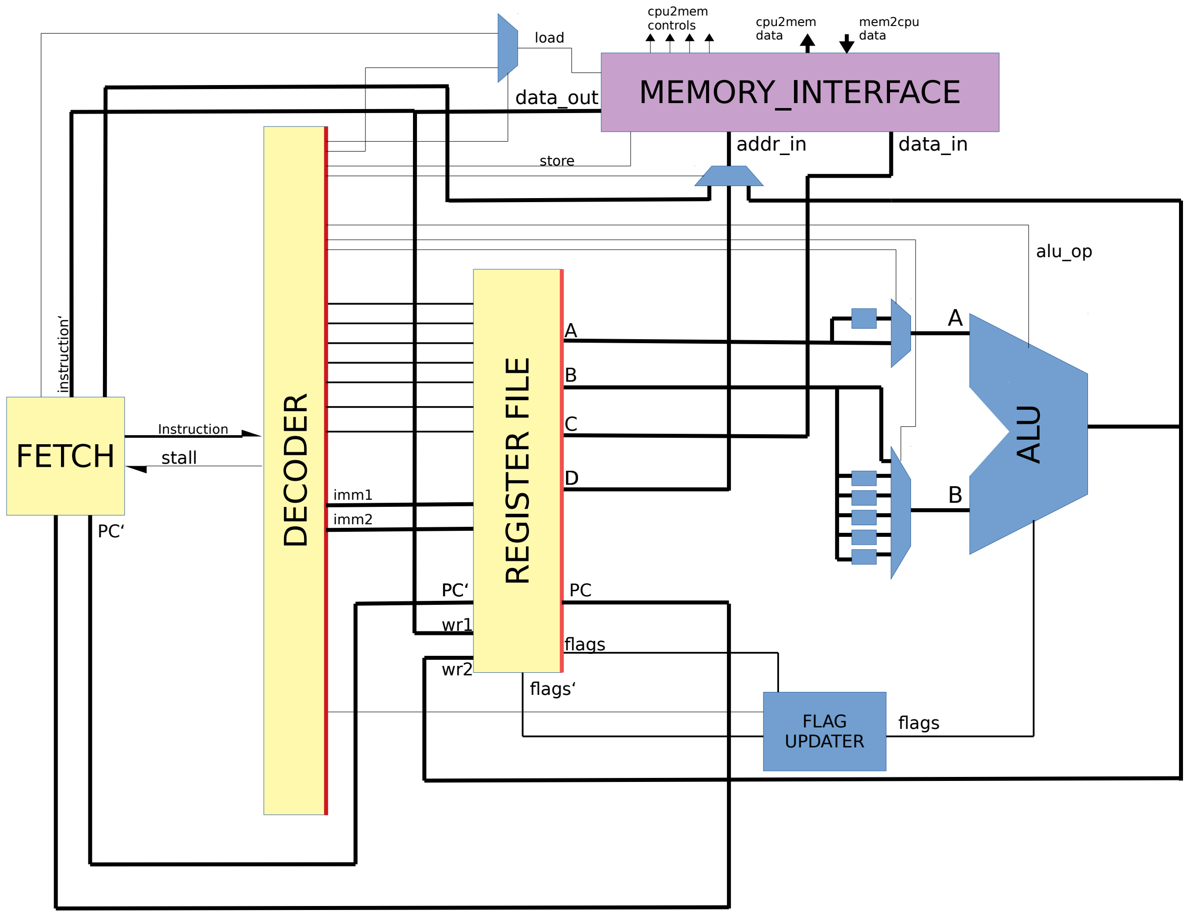

Abbildung 1: Overview of the designed micro-architecture. Pipeline registers are marked red.

1.6 Register File

The register file features four read- and 2 write ports. Additionally, it has separate outputs for the PC and the status flag register.

1.7 Instruction Fetch

Instructions are loaded from the memory by the instruction fetch. It also updates the PC if no stall or branch is applied.

2 Performance

The performance of the designed processor is shown in table 2.

Type	Clock Period Time	max. Frequency	#Cycles/count32	#Cycles/memcp46
Value	1ns	2GHz	300	400