Embedded Systems

mor1kx - Embedded CPU

Write Back Cache Implementation Reference Manual

Francesco Maio, Giovanni Scotti

Document version: 1.0
May 4, 2017

Contents

Section 1

Introduction

1.1 Purpose

The Write Back Cache Implementation Reference Manual for the mor1kx embedded processor is intended to describe how the original architecture has been modified in order to provide the cache's write policy known as *write-back*.

This document aims to supply an adequate knowledge to those developers and programmers, who have to change, adjust and enhance for any reason the related Verilog HDL code, making the understanding of what has been implemented much easier.

1.2 Scope

The mor1kx is an open source, Verilog implementation that is fully compliant with the OpenRisc architecture. In its *Cappuccino* fashion, it offers a six stages pipeline with *write-through* L1 caches.

The following sections deal with all the adjustments required to offer the more aggressive *write-back* writing policy with respect to the *Cappuccino* pipeline only.

1.3 Definitions, Acronyms, Abbreviations

a.k.a: Also Known As.

HDL: Hardware Description Language.

Section 2

Original Functioning

In this section the original operations of the mor1kx, concerning the write policy of the data cache, are briefly described to better understand which Verilog modules need to be modified and why.

2.1 Write Through Implementation

The followings are the relevant features offered before the changeover:

- ***write-through* write policy:** when a write occurs, data is written both to data cache and main memory;
- **no-write allocate:** on a write miss, data is directly written in main memory bypassing the data cache;
- **store buffer:** if enabled, during a store operation data is not only sent to the data cache, but also to the store buffer (a.k.a. write buffer). It is useful because bursts of writes are common. Moreover, it might prevent the processor from stalling.

The flow diagram in Figure 2.1 depicts the series of procedures accomplished by the original implementation of the processor during read and write operations.

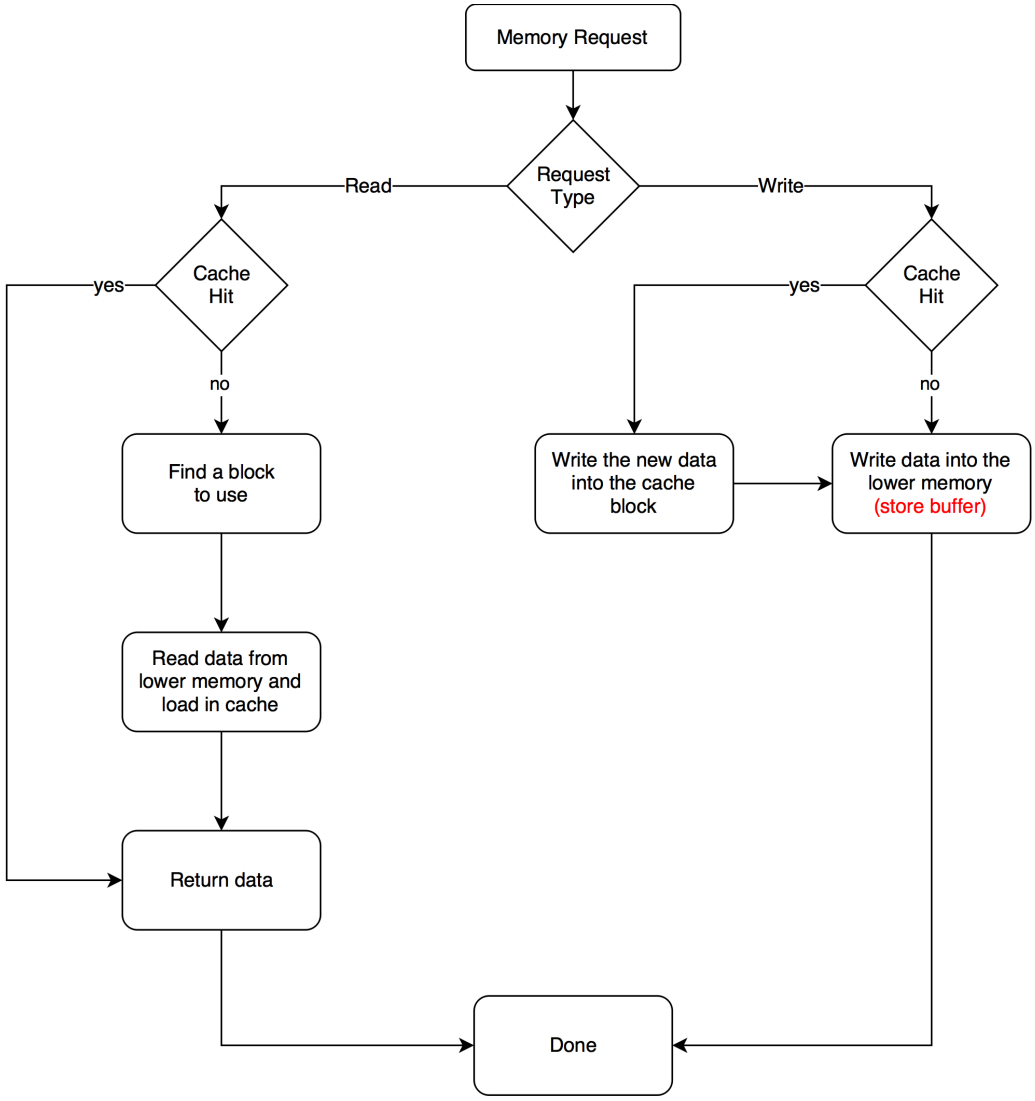

Figure 2.1: Flow diagram showing the original mor1kx *write-through* working principle.

2.2 System Configurations

Different system configurations can be achieved by enabling and disabling the data cache and/or the store buffer. Both the modules are generated in the scope of the LSU module.

Config.	Data Cache	Store Buffer
1	Enabled	Enabled
2	Enabled	None
3	None	Enabled
4	None	None

Table 2.1: Potential system configurations of the original mor1kx implementation. The data cache and the store buffer are generated within the LSU *Cappuccino* module.

Section 3

Write Back Implementation

3.1 Overview