Saarland University
Department 6.2 – Computer Science
Dr. M. A. Hillebrand
Prof. Dr. W. J. Paul

Computer Architecture I – WS 06/07

Exercise Sheet 12 **(due: 05.02.07)**

Note:

This exercise deals with the design of a *write back* cache. A write through cache which you have seen so far, supports the so called *strong memory consistence*: A write always updates the main memory. In contrast to this principle, a write back cache applies the *weak consistency model* which states that a *write hit* only updates the cache. That means that in case a memory line is written which is currently in the cache, is only updated in the cache and not in the memory. This situation will be indicated by a so called *dirty* flag. The main memory is only updated when the whole memory line is written back in case a dirty memory line is evicted to be replaced or a special update request is performed.

This new methodology results in the fact that we can have six 'types' of cache accesses: We can have the following situations: a *hit*, a *clean miss*, and a *dirty miss*, where clean or dirty refers to the fact that the evicted line is not dirty (i.e. clean) or it is dirty, respectively. Each of them can occur for a read as well as for a write access.

Exercise 1: (k-way write back cache) **(8 points)**

Figure 1 depicts the operations of a write back cache for the memory transactions read and write.

Modify the design of the *k-way cache* and the *cache interface* from lecture in order to support the write back policy. Pay special attention to the following aspects:

- A cache line is only considered to be dirty, if the dirty flag is raised and the line holds valid data.
- The memory environment now performs two types of burst accesses: the line fill and the write back of a dirty cache line. The data RAMs of the cache are updated on a line fill but not at a write back.

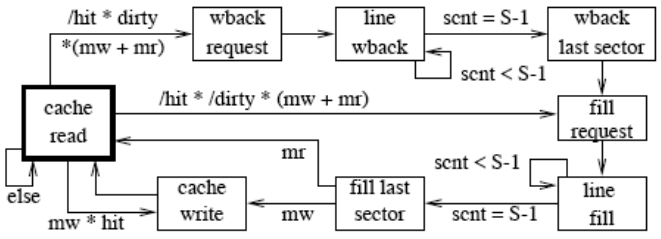

Figure 1: Cache operations of the memory transactions read and write

Exercise 2: (integration into DLX design) **(8 points)**

In class, the integration of a cache into the pipelined DLX machine as a *data memory cache*. Integrate the k-way write back cache into the *data memory environment* of the *pipelined* DLX design. Therefore, you will have to change the data memory environment and the memory interface control. Note, that the control automaton from Figure 1 has to be extended by the bus operations.

Exercise 3: (cache accesses) **(2+6+6=14 points)**

As mentioned before, we have six different cache accesses. On the course web-page in the layouts section, you find the timing diagrams for three of them, namely for a 'read-hit', a 'clean-miss-read', and a 'dirty-miss-write'. Draw the timing diagrams for the remaining 3 cases, namely:

1. 'write-hit'
2. 'clean-miss-write'
3. 'dirty-miss-read'