Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
M. Sc. Christian Müller

Computer Architecture I - WS 07/08

Exercise Sheet 12

Organizational stuff: Please evaluate the lecture until 08.02.08! See
http://frweb.cs.uni-sb.de/03.Studium/08.Eva/index.php?theme=1&lang=en
for more details.

Excercise 1: (fully associative cache)

1. What is the difference between *k-way associative* and *fully associative* cache?
2. What is the main advantage of a fully associative cache wrt. a k-way associative cache?

Excercise 2: (finite state transducers)
Assume for simplicity that our memory system consists of only one direct-mapped cache. Let us concentrate for simplicity only on the data transfer from the main memory to the cache. The data input of our cache is connected directly to the memory output. However, the cache should only 'record' the input data if the memory signalizes its validity according to the protocol defined in the lecture.

In order to communicate with memory over this protocol, our cache system needs a bus control circuit. This bus control circuit bases on a finite state transducer, which changes its state according to the communication progress. This progress is signalized by the handshake signals *reqp* and *brdy*. In each state, the transducer produces a bitvector as output, which has to be interpreted as input signals for the cache as follows:

$$vw \circ val\_in \circ tw \circ cdwb[B-1:0]$$

That is, the bus control circuit should control the cache by computing the correct input signals (except for the address, we don't have to take care about it in this exercise) at a line fill. Basically, before each line fill of address a, the cache line $a.line$ should be invalidated. During the line fill, the write enable signals *cdwb* should be set only if correct data (signalized by *brdy*) is on the memory output. After all sectors are transmitted (each sector consists of B bytes), the cache line $a.line$ should be marked valid and the tag $a.tag$ should be written into the corresponding RAM.

In this exercise you have to construct and implement the finite state transducer of the bus control circuit. You probably will need a detail description of the memory protocol, which you will find in the appendix.
Hint: In contrast to the exercise description, the solution is very simple ;-). The finite state transducer has only 4 states (you can have more, but 4 is enough).

Excercise 3: (multi-level paging)
Ensure that you understand walks and walk extension and can define them.

Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
M. Sc. Christian Müller

Computer Architecture I - WS 07/08

Exercise Sheet 12

Appendix

The communication of the cache and the main memory can be described as follows. The bus control raises the signal *req* in order to indicate to the main memory a new access and the signal *mw* for a write access. The main memory signals a pending request by raising the signal *reqp* in the next cycle. This signal stays active until the last data word is acknowledged. Ready data are signaled by the memory *one cycle in advance* by raising the acknowledgement signal *brdy*. That is, at a read access, one cycle after *brdy*, the data on the memory output bus is guaranteed to be valid.