Saarland University
Department 6.2 – Computer Science
Prof. Dr. W. J. Paul
M. Sc. Petro Lutsyk

Computer Architecture – WS14/15

Exercise Sheet 12 (due: 03.02.15, 40 points)

Exercise 1: (correctness proof) (12 points)

Recall the last exercise sheet, where in the last problem you were asked to change generation of the control signals, s.t.

$$idle(i)^t \rightarrow \neg grant[i]^t.$$

One way to achieve this is to lower the bus request signal cycle earlier, on transition $mdata \rightarrow w$. However, this modification is not sufficient, since (in the implementation) it can happen that another master j waits to go global in cycle t (state $wait(j)^t$), while its slave is updating the same line. In case if the state changes (imagine j was an owner of the shared line), if j goes global cycle earlier, it will put wrong protocol signals on the bus. Though, this technical issue can be easily repaired, namely by forwarding from the input b of the state RAM to the input of the auxiliary register $souta'$.

Find the **first** lemma (after *"auxiliary registers"*) which proof breaks because of these modifications. Give the exact line from the proof where the argument fails. Try to fix it locally, i.e., complete the proof from the place where it breaks without further modifications of the design.

Exercise 2: (bus contention) (4 points)

Using your lecture notes (or the corresponding sections of the script) answer the following questions.

(a) In the lemma *"no contention 2"* from the book, the absence of bus contention is established for all signals X on the tristate bus except for

$$X = bdout.$$

Explain (in words) why exactly we can not establish the result for *bdout* as well.

(b) In the lemma *"no glitches, disabled"* from the book, we forbid turning off those drivers which are already disabled. However, we do not forbid to clock the data registers of these drivers. Explain (in words) why exactly we do not forbid to clock these registers as well.

Exercise 3: (timing diagram) (12 points)

In this exercise you need to draw an approximate (without minimal and maximal propagation delays) timing diagram which illustrates the following transaction.

The master of transaction – processor i[1] – performs a global write access to the line at address a, which is present in its cache as shared data, owned by some other cache j. For every access below, depict master and slave automata states (like in the formulation of lemma *"sync"* in the lecture), protocol and data signals on the bus, clock-enable signals of registers, and control signals of drivers.

- global write by cache i on line address a
- delayed local by cache j (owner) on line address a
- local read by cache k on line address a
- local write by cache l on line address b

Assume that initially (cycle t_0) all caches mentioned above are in state *idle*, cache i is granted first.

Exercise 4: (local order) (12 points)

Prove the corresponding lemma from the book.

[1] assume every processor has only one cache