Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
Geng Chen

Multicore System Architecture - WS15/16

Exercise Sheet 7 (due: Dec 11, 2015)

Important:

- Each week on Friday one exercise sheet will be released. The solutions should be handed in before or after the next Friday lecture if not stated otherwise. For the admission to the exam you will need at least 50% of the points of the exercises.
- Tutorials: Group A: Wednesday 12:00 - 14:00 (Room 328, E 1 3, Tutor: Jonas Donia) and Group B: Tuesday 10:00 - 12:00 (Room 328, E 1 3, Tutor: Shahd Zahran)
- You are allowed to solve the exercise sheets in groups. Everybody who has his or her name on the solution must be able to present it in the tutorials. Everybody must present the solution of at least two exercises.
- Please, register for the lecture at the lecture's webpage until Nov 13th, 2015!
  http://www-wjp.cs.uni-saarland.de/lehre/vorlesung/rechnerarchitektur/ws15/anmeldung.php
  Also do not forget to register for the exam in the HISPOS system!
- The oral exam will take place in February. An exact date will be decided upon in class.

Tutor: _____

Name, Matr. Number: _____

Exercise 1: **(8)**

Recall the *data paths* for the data, tag and state RAMs used in the construction of the cache system. Using the lecture notes (or the corresponding sections of the script) repeat the following arguments.

1. Explain what we need the auxiliary registers (e.g. $dataouta'$, $tagouta'$, $souta'$ and $soutb'$) for.
2. Explain why we use the register $souta'$ on transition $wait \to m0$, and why not on $flush \to m0$.
3. Explain why we forward state I to the inputs of circuit $C1$.
4. Explain why we forward state M to the inputs of circuit $C2$.

Exercise 2: **(4)**

To allow interrupts in system mode, we need to add a special purpose register with the index 9 and the synonym *emode*. The $spr(emode)$ saves the $spr(mode)$ when interrupts happen and restore the *mode* register when returning from an interrupt handler. Extend the semantics of $jisr$ and *eret* by adding the *emode*.

Exercise 3: **(4)**

Construct an SPR-RAM which has

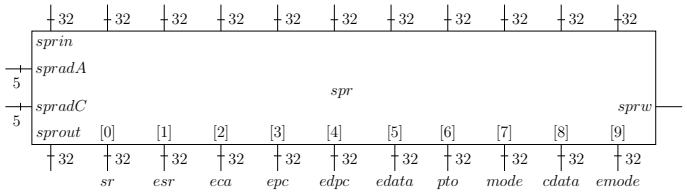
- a 32-bit data input $sprin$,
- a 32-bit data output $sprout$,
- a 5-bit read address $spradA$,
- a 5-bit write address $spradC$,

Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
Geng Chen

Multicore System Architecture - WS15/16

Exercise Sheet 7 (due: Dec 11, 2015)

- a write signal $sprw$,
- for each $i \in [0:9]$. individual 32-bit data input and output for $spr(i_5)$.



And prove the correctness of your design.

Exercise 4: **(2)**

Construction data paths between general purpose registers and general purpose registers for the support of $movg2s$ and $movs2g$ instructions. (1 point) Extend the definition of Cad. (1 point)

Exercise 5: **(4)**

Show the correctness of instruction fetch (input of stage 1) in the multicore processor correctness proof.