

Computer Architecture I - WS 07/08  
Exercise Sheet 10

---

**Exercise 1: (warm up)**

1. Construct a single port RAM  $R$  with inputs  $ad \in \mathbb{B}^a$ ,  $din \in \mathbb{B}^n$ ,  $w \in \mathbb{B}$ , and  $clear \in \mathbb{B}$ . The only output is  $dout \in \mathbb{B}^n$ . The RAM should be a standard RAM except for the fact that the clear input clears every address, i.e. in case  $clear^t = 1$  we have that  $R^{t+1}(ad) = 0^n$  for all  $ad \in \mathbb{B}^a$ .
2. Construct a circuit with inputs  $d \in \mathbb{B}^2$  and  $ea \in \mathbb{B}^2$ , which can be used for misalignment interrupt computation. You should use the access width bits  $d(c)$  and two last bits of the effective address  $ea(c)[1 : 0]$ .

**Exercise 2: (selfmodifying code)**

1. Explain the main architectural problem we have with a selfmodifying code (wrt. pipelining).
2. In class, you have seen the software convention that between an instruction that writes to some address  $a$  and an instruction that reads from  $a$  there has to be an instruction that empties the pipeline. Formally:

$$\begin{aligned} \forall i, j, a : i < j \wedge (I_i \text{ writes to } m(a)) \wedge (I_j \text{ is fetched from } m(a)) \\ \Rightarrow \exists k : i < k < j \wedge (I_k \text{ empties the pipe}) \end{aligned}$$

where  $a$  is an address in the code region. In this exercise, you have to implement a mechanism that makes the rfe instruction such a draining instruction.

3. Assume that the software convention from the previous exercise does not hold. Then, one has to speculate that the convention is fulfilled. Hence, in case it is not fulfilled, we have to rollback the computation and draining the pipe. Implement a circuit that checks for misspeculation, performs a rollback in case of a misspeculation, and drains the pipe.
4. One could think, if we forbid to users to execute selfmodifying programs we could avoid the problem mentioned in exercise 2.1. However, in class you have seen an example, where the operating system together with a user program turns into a selfmodifying code at page swapping. Find another example for such behaviour of the operating system caused by user activity.

**Exercise 3: (introduction to memory system)**

1. Explain the write strategies *write through* and *write back*.
2. What is snooping?