

Jan Reineke  
Andreas Abel



---

Deadline: Thursday, July 4, 2013, 14:15

## Assignment 9

### Problem 1: Questions (3+3+3+4+4+4+4+5 Points)

1. Assume we want to determine the worst-case execution time of a program. Why is it not a good idea, in order to simplify the problem, to assume that all memory accesses result in cache misses?
2. We know that a cache miss takes more time than a cache hit. Assume that a *must analysis* is not able to predict that a certain memory access always leads to a cache hit. Is it then safe to assume that this access always leads to a cache miss in order to compute the WCET of a program?
3. Why is LRU cache replacement a better choice for predictable architectures than FIFO?
4. We have seen in class that the worst-case execution time of a program depends on the state of the cache at the start of the program. Can we compute an upper bound on the WCET by assuming that the cache is empty initially? What properties does the processor need to have for this to be a correct approach?
5. Assume we have a cache with associativity 8 that uses the FIFO replacement policy. Further assume that we are given two different initial cache states  $s_1$  and  $s_2$ . What is the maximum length of an access sequence such that the last access can lead to a cache hit for  $s_1$  and to cache miss for  $s_2$ . Can you find an example to justify your answer?
6. Explain what is meant by the “state explosion problem” with respect to timing anomalies.
7. Is it, in general, safe to analyze the WCET for different components of the microarchitecture (like caches, pipelines, or branch predictors) separately, and then add up the individual results to compute the total WCET?
8. How can other processes affect the execution time of a program (a) on a single core, (b) on multi-cores? Note that we only consider the *execution time*, i.e., the time the program is actually running, and not the *response time*. What are mechanisms to eliminate such interference?