SAARLAND UNIVERSITY
COMPUTER SCIENCE

Prof. Dr. Jan Reineke
Sebastian Hahn, M.Sc.

Verification of Real-Time Systems SS 2015

Assignment 11

Deadline: July 16, 2015, before the lecture

Exercise 11.1: Questions (1+1+1+2+2+2+1+2+1+2+1=16 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 4 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. Find an example to justify your answer.

(6) Demonstrate (if any) a domino effect for PLRU caches of associativity 4 and 2. Justify your answer.

(7) Explain what is meant by the "state explosion problem" with respect to timing anomalies.

(8) Explain why it is expensive to use cumulative information in a non-compositional fashion.

(9) 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?

(10) 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? Name at least two.

(11) Why can it be favorable to use shared resources with temporal isolation instead of corresponding private resources.

Exercise 11.2: Timing Anomalies (2+2+2=6 Points)

(1) Find scheduling anomalies for the following cases: (a) by removing dependencies on a system with two identical resources, (b) by increasing the amount of identical resources from two to three.

(2) Find a scenario similar to scheduling anomalies, but such that the local worst case (p cycles longer than the local best case) leads to a larger increase ($> p$ cycles) in the overall execution time on a system with two identical resources. Explicitly mark dependencies and when individual tasks become ready.

(3) How do scenarios as described in (2) affect timing compositionality?

Exercise 11.3: Predictability (1.5+3+2.5=7 Points)

Three important properties of predictable architectures are *temporal isolation*, *timing compositionality*, and the absence of *timing anomalies*.

(1) Briefly explain each of these properties.

(2) Describe possible techniques to achieve these properties. Why are these techniques often not used in common architectures, i.e., what are their drawbacks?

(3) Which of the properties does the PRET PTARM architecture achieve, and by which means? Name two drawbacks.