

UG Exam Performance Feedback

Second Year

2016/2017 Semester 2

COMP25212 System Architecture

Javier Navaridas-Palma
Antoni Pop

Comments Feedback for B3/B4:

The students seem to have grasped well the main concepts related to pipelining (Q10). The main issues here have been not being able to quantify the penalties induced by the different hazards, which transforms the question from practical to bookwork and not quantifying properly. In most cases of the latter, the quantification was off by one cycle but the main idea was explained well. In these cases, the final marks have only been penalised slightly. Non-justified values that were incorrect, haven't been considered, though. One common misunderstanding of students is that branch prediction is always good for performance.

While it normally is, and that's why it is implemented in most processors, in some pathological cases it might not be, e.g. if the prediction is always wrong, it'll be counter-productive when compared with doing nothing and just by chance not taking some of the branches.

This hasn't been penalised, but has been common enough so to mention it here.

The question about scalability limits (performance walls) has obtained poorer marks as many answers have been rather unfocused. The ILP wall (lack of parallelism within applications) is the one that has been missed more often, while memory and power have been more commonly identified, although quite often not explained in enough detail.

The pros and cons of reordering instructions in the compiler and hardware (Q11) seem to have been understood well, with not too many issues. The most common error here has been to discuss the positives of reordering without doing the comparison between compiler and HW.

The question about deciding the best processor for a robot-control system (Q12) which shows the understanding of the different mechanisms that can be implemented in a processor has been answered generally well.

As in Q10, the main issue here has been to just dump details of the different architectures from memory, without actually relating it to the use case. A common oversight from students is the overheads of cache coherency for multicore and/or the exploitation of cache locality for multithreading, which should give an edge to the later.

Q13 and Q14 results show that most students haven't understood properly the concept of consistency and have defined it as some subset of cache coherency. The discussion on synchronization mechanisms for consistency is generally very poor and, again, most students have discussed coherency-related stuff. Given data consistency and synchronization are really important concepts in computer science, I suggest students should revise this part of the curriculum. Considering these are aspects that have been covered already in 25111, I was expecting a much better outcome in these two questions.
