The exam was generally answered very well this year. The overall average mark was 66.38%.

For some reason, question 1 was not popular; fewer than 10% of the class chose to answer it. This meant that almost everyone answered questions 2, 3 and 4.

Question 1 had the lowest average mark (9.33/20), but this appears to be due to it having been chosen predominantly by students who did not do well in the exam. The average overall mark of the 6 students who answered question 1 was in fact slightly less than the average they achieved for the question. With so few answers there is little to say about trends for answers to this question.

Question 2 had the second highest average mark (13.79/20), indicating that it was generally very well answered. Marks that were lost came from every part of the question, either for missing out important detail, or for not understanding how instructions like ‘load linked’ and ‘store conditional’ work.

Question 3 had the second lowest average mark (12.54/20), but was still generally answered very well. Several candidates lost marks because they were unable to generalise the MSI coherence scheme to include the E state. Similarly, marks were lost in part (c) for not accurately keeping track of the cache line state changes in the 3 cores; surprisingly, more than a few candidates thought that the cache line state would change when an add instruction (which operates solely on the registers) is executed.

Question 4 had the highest average mark (13.86/20) and was thus the best answered question. For many candidates, this was the last question they answered, so the high average mark seems to indicate that students mostly had time to complete their answers to the paper. A significant number of answers to part (b) gave either no locking code or code that did not do the necessary job. In fact the code simply needs to take 2 locks, one for element a[i] and one for element a[j], do the swap and then release the 2 locks. The code can suffer from deadlock if 2 threads execute it with the values of i and j in the opposite sense to one another. No-one spotted that it would also suffer if the values of i and j were equal! For part (d), few candidates mentioned that eager validation is best employed when transactions are likely to execute for long periods of time (worth 1 mark).