All individual questions got fully (or at least close to fully) answered by at least one student (typically many more). However, there was quite a variance in the overall performance. All asked questions were very closely related to the course material and the lab assignments. However, answers giving variants from the course material got marks, as long as answers were reasonable.

Question 1 "Questions concerning the ARM architecture" was mostly well answered and only a few students couldn't describe the impact of compiling for ARM 32-bit versus Thumb, when a system is once memory bound (Thumb is better) or when we are concerned about best performance in a memory unbound system (ARM 32-bit is better)

Question 2 "Questions concerning programming the ARM architecture" was not as well answered as the others. Many students were not able to initialize an array with a constant, which should be easy, when done already Project A.
The question "Describe the steps that have to be carried out by a (simple) operating system running on an ARM core to swap between two tasks ..." was in most cases not well answered. First of all, the question asked what "you as a programmer / or the OS" has to do, it was not what the ARM does when it receives an IRQ. The question was intended to be easy for all students who did the context switching exercise before. So the expected answer was about where to store the context (stack is not a good ideal), how to get the state (through system mode), how to get the proper PC value for resuming a task, how to store/restore the status register (MRS & MSR). Then we have to do something that we know which task is to be executed next (any sensible answer would be accepted here), Deal with the interrupt, and so forth.

Question 3 "Questions about memory, memory management, and cache" was mostly well answered. For the answer on cache power consumption versus cache organization, answers like increasing cache line size or playing with associativity (for having less memory accesses) were accepted as long as this was somehow explained. Some students had problems to answer "How big is the page table (in terms of bits)?". Well, with 20 bits connected to the table, it is 220 x 20 bit = 20 Mbit. 220 x 32 bit was also accepted, as it sensible to assume that the table is stored somehow 32-bit aligned.

For the Question "Sketch the basic organisation of a Protection Unit for 8 memory regions as used in ARM and describe the principles of its operation?" either the figure showing the architecture of the protection unit or the figure showing the different address regions was accepted here.

Comments

All individual questions got fully (or at least close to fully) answered by at least one student (typically many more). However, there was quite a variance in the overall performance. All asked questions were very closely related to the course material and the lab assignments. However, answers giving variants from the course material got marks, as long as answers were reasonable.

Question 1 "Questions concerning the ARM architecture" was mostly well answered and only a few students couldn't describe the impact of compiling for ARM 32-bit versus Thumb, when a system is once memory bound (Thumb is better) or when we are concerned about best performance in a memory unbound system (ARM 32-bit is better)

Question 2 "Questions concerning programming the ARM architecture" was not as well answered as the others. Many students were not able to initialize an array with a constant, which should be easy, when done already Project A.
The question "Describe the steps that have to be carried out by a (simple) operating system running on an ARM core to swap between two tasks ..." was in most cases not well answered. First of all, the question asked what "you as a programmer / or the OS" has to do, it was not what the ARM does when it receives an IRQ. The question was intended to be easy for all students who did the context switching exercise before. So the expected answer was about where to store the context (stack is not a good ideal), how to get the state (through system mode), how to get the proper PC value for resuming a task, how to store/restore the status register (MRS & MSR). Then we have to do something that we know which task is to be executed next (any sensible answer would be accepted here), Deal with the interrupt, and so forth.

Question 3 "Questions about memory, memory management, and cache" was mostly well answered. For the answer on cache power consumption versus cache organization, answers like increasing cache line size or playing with associativity (for having less memory accesses) were accepted as long as this was somehow explained. Some students had problems to answer "How big is the page table (in terms of bits)?". Well, with 20 bits connected to the table, it is 220 x 20 bit = 20 Mbit. 220 x 32 bit was also accepted, as it sensible to assume that the table is stored somehow 32-bit aligned.

For the Question "Sketch the basic organisation of a Protection Unit for 8 memory regions as used in ARM and describe the principles of its operation?" either the figure showing the architecture of the protection unit or the figure showing the different address regions was accepted here.