Exam Feedback for Systems and Networks May 2018

Q1

(a) is an application of Section 2 of the notes.

(b) The correct answer here is: “it depends on what code is being used”. If the programmer is using an unsigned code, this will overflow; if a two’s complement code, it will not.

(c) The easy way to copy R1 into R2 is just: ADD R2,R1,R0

(d) First determine whether the most significant bit (MSB) of R2 is a 0 or a 1. Then logic shift R2. If the MSB was a 0 do nothing; if it was a 1, add 1.

(e) This is very straightforward. Use the same technique as in the coursework exercise to determine whether array element X[i] is odd or not (either use DIV or AND as you prefer) and store the result in Y[i].

(f) This is a simple modification of A. Set up a sum variable in a free register. Each time round the loop add the value of Y[i] into that register. Then, at the end, do an odd/even test on the final sum and store the result in PARITY.

Q2

(c)-(g) are based on the material on caches in Section 10 and is patterned after the example on Slide 5. The program looks like an attempt to add two arrays, but it contains four bugs. The question was marked very openly and students who identified bugs were given full credit, whether they removed them or not, if they subsequently answered correctly.

The bugs are as follows. Firstly R1 is initialised to hold the increment constant 1 but is then overwritten with X[0] which is given as 0. Secondly R2 is used as the array index but is overwritten on each cycle with Y[0]: the value of Y[0] is not given and is irrelevant. Thirdly, R0 is used instead of the index, R2, to access X[i] and Y[i]. Finally, the condition for the loop CMPGT would perform an extra cycle even if R2 were being used correctly as it would allow the loop to range from 0 to n instead of 0 to n-1.

(c) Each time round the loop the program adds the same two values X[0] and Y[0]. R2 is loaded with Y[0] but is then copied into R3 by ADD R3,R1,R2 (remembering that R1 is 0). Since R2 and R3 are the same, the CMPGT will never be true and so the program will loop for ever. There is no need to know the value of Y[0] and this is not given.

(d) This is straightforward (1+2+3+3+1+3+1+2 = 16 cycles)

(e) The program will never terminate.

(f) After the first time round the loop, all accesses except the STORE data write will be a cache hit (written data always has to be recorded in memory so a write can never be a cache hit). So, there will be 15 cache hits every loop.

(g) The program will loop forever but loops after the first one will be faster (25ns instead of 160ns).

Q3

(a) is from Section 12, Slide 3.

(b) Here you are expected to reason from your understanding of how a broadcasting system works.

(i) One to many communications is easier in a broadcast network because every message goes to every host. By contrast, in a non-broadcast (switched) network, one-many communications is more awkward since messages by default are directed to just one destination. here nodes would typically need to replicate messages they had received on several outgoing links.

(ii) Security is harder in broadcast networks because everything attached to the network sees all the messages. In a switched network, a normal message travels along a restricted path and is much harder to intercept.

(iii) In a broadcast network whenever any host is sending it takes up the whole medium to the exclusion of other aspiring senders. These other nodes that want to send must wait to get a turn. The more hosts there are, the longer any single host will have to wait on average. Beyond a certain size, hosts would have to wait so long to get access to the medium that the performance would be unacceptable.

(c) is from Chapter 9 Slide 9 (and preceding)

(d) is from Section 12 Slides 8 and 11. The transport layer uses different ports to support different concurrent client or server applications engaged in separate communications with remote clients or servers.