Please see PDF for feedback.
COMP2211 2011/12 Exam Feedback

Question 1 (PWN)

On the whole the question was answered very well.

a) On the whole answered correctly. A number of answers stated that the reason why the STUMP has only 8 instructions is because it is a RISC processor. Yes, a RISC processor has a limited instruction set but the reason is that we only have 3-bits in the instruction to specify the type of instruction, which only allows a maximum of 8.

Some answers did not give code examples of how to implement MOV and NOP as requested. A number of answers discussed the use of load/store for implementing a MOV. A MOV can be implemented simply using ADD where one of the source registers is R0, which is always zero.

b) Some mixed answers to this question, particular problems were:

- The code is given with the status of the processor at the instruction at line 4. In some scripts answers were generated by evaluating the code from line 3 again – this resulted in incorrect answers because of the duplication of the instruction at line 3.
- The instruction at line 9 is not executed due to the branch in the previous instruction occurring.
- Some answers did not recognize that R0 is hard wired to 0.
- Parts ii) and v) resulted in the status flags being set. Some answers didn’t recognize that the status register is set, or they simply gave an incorrect binary value for the status register.

c) Almost everyone answered this question correctly.

d) On the whole Ok. A mark was awarded for suitable diagrams, which were missing in a number of answers.

e) On the whole answered well.

Average mark: 67%.

Question 2 (PWN)

One the whole a well answered question.

a) I was looking for a description for the following four functional blocks:

Register bank (including mention of the Program Counter)
Execute unit (including mention of Result Register)
Data Interface (including mention of Instruction Register)
Address Interface

b) This was answered well. A simple description of the 4 types of instructions was required including the following information:

- Type 1 – I expected the discussion to include the shifting of data from A when required.
- Type 2 - I expected the discussion to include the need for sign extension of the immediate value
- load/store – can be Type 1 or Type 2 AND it is an address that is calculated (a large
number of answers failed to identify this)
•branch – offset from instruction sign extended and added to PC value and only written back
  to the PC if the branch is to be taken (I wasn’t bothered about the need to take into account
  that the PC had been incremented in the fetch phase)

I did not request detail about how the instruction is formed for each type of instruction.
Simply how the instruction relates to the RTL datapath.

c) A discussion of

•reg – this does not necessarily translate into a register, it will be a variable that retains its
  value until reassigned.
•always – need to mention sensitivity list, in the example code given the always block is
  executed on a positive edge of the signal clock
•<= - non-blocking assignment, multiple assignments performed in parallel. Control
  statements, if .. else, used to control behaviour.

Average mark: 67%.

Question 3 (JDG)

This was clearly an unpopular question; few candidates attempted it. Of those who did, half
understood the principles and got good marks, the others clearly did not and the marks were
very poor. This is somewhat disappointing in that, in principle, the series/parallel constructs
in a CMOS gate should be quite simple to understand and the example is a common one in
many VLSI texts.

There is little point in giving a section-by-section breakdown except, perhaps mentioning on
part d that the first factor of two was typically grasped but the series transistors needed to be
inherited from the earlier parts for full marks. Even the trivially simple arithmetic put some
people off!

Question 4 (JDG)

This question was largely book-work and should have been quite simple. There appears to
be problems with understanding on what is going on in some places.

a) It might be expected that the idea of regression testing was familiar in a wider context.
   Sadly only about half the candidates knew clearly why this is important.

b) This was recognised and answered sensibly by most.

c) Very few people appear to have read the question! It asks why the test patterns might
differ. A typical answer is that 'testing is important' and, maybe, 'designs can be tested by
simulation'. The object was to think what the tests might be intended to discover and how
best to achieve that.

d) Most people got the sketch 'more or less' the right shape. Again, if the principle is
understood then the mechanism should become fairly obvious. The evidence points to
repetition of a (half) remembered picture without the understanding of what it means.

A few candidates discussed timing and parametric measurements here; again this points to
grasping at keywords rather than concepts. It is difficult to see how scanning logic blocks
might help with parametric measurements.

e) In general this was well known.

F) This was intended to be difficult (although some suggestions were mentioned in a
lecture). There was some occasional inventive thinking which was credited when
appropriate.
Section A:

This was the mandatory section covering all of the delivered material in the form of multiple choice questions. This section was well answered with an average of 75% of questions being correctly answered, especially parts related to the syntax of SQL, the semantics of the relational algebra expression, and concepts of the relational model.

Section B (JAK)

Summary: About two-thirds of students answered this question. Generally Part A was very well answered; Part B was well answered; and Part C had some very good reflective answers but fewer than in the earlier parts.

Q1
a) EER diagram
   i) Diagram
   Very well answered overall.

   ii) Relational schema
   Very well answered overall.

B) Normalisation
   i) Inference rules
   Generally well answered
   ii) 1NF
   Generally well answered
   iii) 2NF
   Generally well answered
   iv) 3NF
   Reasonable: some good, some struggled.

C) Discussion: relational schema in terms of ER diagrams
   This involved some reflection about the purpose of the ER/schema process. Some very good answers, and most who gave substantial answers did well.

Section C: (SS)

This was the least popular section, covering mainly the topics of File Organisation and Indexes. Despite the fact that only 38% of the students chose to solve this section, the average mark for it was reasonably high.

In general, for this section, bookwork and application types of questions were well answered. However, original thought questions were less than well answered, showing, in a few cases, poor analytical skills.
Comments:

Question 1: MCQ - restricted

Question 2: (RDS)

In general, this question on drawing an activity diagram and giving use cases was not answered well, though there were some excellent answers.

The activity diagram in part A was harder than it might initially appear, though some exam technique would have helped and remembering of your own exam technique would have helped. I asked for an activity diagram that captured the process of setting and sitting an exam. I gave an outline of the setting and marking of an exam; you sit exams, so you should know this bit. A good answer would have taken the elements of setting, writing rubric, moderating etc (given in the question). And mixed this with your own experience of reading the paper, choosing which questions to answer, checking the answer and so on. This would have satisfied the need to explain exam technique to another student. Many answers just missed out the detail on sitting an exam, but included extraneous detail on attending the course and such like.

Part b on defining an actor was a simple knowledge recall exercise; I gave the answer in the lecture notes and workshop handouts. Mainly partial or erroneous answers were given.

Part c on getting use cases and actors was better than Part a for this question. Many use cases were identified well, but there was some diving down into too much fine detail. Non-human actors were often missed out.

Question 3. (KKL)

Overall, the answers were good, showing a good understanding of the course material.

(a)
(i) Good answers in general. Some confusion between domain classes and system classes. Some answers are too simple, e.g. "structural model defines the structure of the system"; need to explain domain and system classes.
(ii) Good answers in general. Some answers are too simple, e.g. "behavioural model defines the behaviour of the system"; need to explain interactions between system objects, and the kinds of diagrams used to define them.

(b)
(i) Good answers in general. Some confusion between domain classes (only relations) and activity diagrams (flow of activity).
(ii) Good answers in general. Some confusion between domain classes and system classes; unusual for system classes to be the same as domain classes, e.g. a system interface is typically added to refined domain classes (these usually do not include a system interface).
(iii) Good answers in general. Some confusion sequence diagrams (interactions between system objects) and activity diagrams (business process).
Ke Chen:-

This report is regarding questions (6)-(10) in Sect. A and those in Sect. C.

Regarding short questions (6)-(10) in Sect. A, around 40% students achieve the satisfactory performance while others do not seem to gain the essential understanding for book knowledge that had been highlighted and repeated many times during lectures, revision lecture notes and non-assessed exercises. For example, a number of students could not point out the essential difference between supervised and unsupervised learning. Even though a mini-project was done and the same question was asked in exercises, around 50% students still could not tell the underlying difference between Bayesian and Naïve Bayesian classifiers precisely even after they have already done a highly relevant Lab exercise.

For two comprehensive questions, i.e., questions C1 and C2, in Sect. C, around 20% students well understood the problems and applied appropriate knowledge and critical analysis skills to those difficult problems (although most of their answers were less perfect) while a number of students could not answer those essential questions in either C1 or C2. Given the fact that most of students chose C1 instead of C2, the main problem appeared in C1(c) and C1(d) where few students fully understood the cost function for SVM learning as it is expressed in a formal way as well as what “kernel trick” means in terms of computationally efficiency.

In summary, the overall result reflects what students actually achieved from this course unit. In comparison (with the same syllabus), students this year seems to underperformed those last two year in both Sections A (Questions 6-10) and C.

Gavin Brown:-

Many students had trouble with the boosting/bagging questions, mixing the concepts up with those of feature selection, filters/wrappers. Boosting and Bagging are nothing to do with the filter/wrapper method, but are ensemble techniques.

The decision tree section was completed well by most, except the final (advanced) question about relating filter methods to splitting criteria - the answer was that they can both use mutual information.

Pete Jinks:-

Please see attached pdf.
Question 1. Algorithm Design

Most students answered this question (75%) and the results on the whole were good to very good, with an average of 65% and a number of students gaining 100%.

Marks were awarded for precise descriptions of correct algorithms, the correct complexities AND your calculations, and presentation of efficient algorithms (25% of the marks went to efficiency). All problems had simple $O(N^2)$ algorithms, more efficient ones were available through various techniques including, sorting, hashing, efficient data structures and combinations of these. Most students presented correct answers for most parts, but there are some subtle features especially of the 1st and 3rd parts. Even with correct answers, some students lost marks through imprecise descriptions of algorithms, and not showing your working in calculating complexities.

Question 2 - Sorting

a) Many answered that an essential property of sorting algorithm is that it MUST be stable, and so answered incorrectly. Others gave a property of the input (e.g. it must be from a total order). The correct, simple answer is that the numbers in the output must be the same as those in the input (just permuted).

b) Most got the properties of Quicksort right. A common mistake was thinking it a distribution-based sort (property A).

c) Most came up with an efficient method for sorting a list of binary digits. Several answers are possible. Some did not make explicit the modification and so lost a mark.

d) The large majority seemed to understand stable sorting correctly.

e) Most were able to recognise merge sort from the list of numbers "nearly sorted". A few thought it was quick-sort, which is not possible. Overall, this question was answered very well.

f) This question about a pivot method in Quicksort was not answered very well. More than half got the first bit right, but the rest did not follow the instruction of how to calculate the pivot (a bit disappointing). The second part seemed to cause confusion, and many only got half marks or zero. Again, it was mainly about following instructions and thinking carefully - some seemed to guess and did not explain what they did.

g) Answered well. Most understood how to re-write the comparison function to get the order specified. Some failed to get some cases right, notably when both a and b were equal to either 0 or 5.

h) The answer to this question about lower bounds on comparison-based sorting was covered in the text book, but very few used the right method. The median mark for this part was 0/3. Some marks were awarded for those who got to the right answer by arguing from merge sort, though this is not quite a valid demonstration that the "minimum" number of comparisons has been found.

Question 3 - Complexity

a) The worst-case complexity of the code fragments was mostly calculated well. Marks were picked up for good working even if the final answer was not right.

B) Most simplified the expression further correctly, but a few seemed to just add a log for no reason.

C) Understanding about a property of log complexities is examined. Some people worked this out (which was simple to do). Some knew the answer. Some guessed and got it wrong. Less than 50% got both marks.
D) About two-thirds got this calculation right.

E) (i) Again, about two thirds did the simple calculation to work out linear complexity that fits the data.
(ii) Most gave two out of three properly separate ways in which data collection could have been improved. Some gave three.
(iii) Quite a large number lost marks because they didn't explain HOW to calculate complexity from the data - or just said "plot a graph" which is obviously too concise for 3 marks. The standard answer is by explaining how to do a log-log plot, check for a polynomial growth and obtain the power and coefficient. But I also accepted sufficiently clear statements about using a computer program to do the fitting, or other explanations about fitting polynomials from tabulated data. One said that it would be possible to use Newton interpolation and got the marks also.
Q1) AFC
a) Answers failed to identify that cannot simultaneously updated all instances of a network application because these instances are distributed in many locations. Others did not make clear that multiple versions have to exist because extensibility means extending functionality.

B) The question asked for a solution for a particular scenario, too many answers gave generic solutions that were not customised to the scenario, whereas others gave examples without giving a definition of the format being used. The question asked for a design of messages and the messages exchanged, often no structure for messages was given. The scenario required the searching nd the return of results, some answers only addressed the searching issue. Answers often had complicated mappings to allow the transfer of values for fields; a simple smtp/HTTP key/value pair approach would suffice.

c) Answer failed to address the why incorrectly interpreted data can occur. They also failed to give the examples requested in the question text.

d) Solutions needed to address authentication and detecting messages that had been tampered with, many only did one of these. Too many answers failed to use the client id to allow the server the support multiple users. They assumed that the server magically knew the correct key for a client or that all clients shared the same key. As part of this problem, some answers encrypted the client id, which leaves the server no way of determine the correct key the decrypt the client id. Many answers failed address the minimising reverse engineering part of the question; this was just looking for the use establishment and use of a symmetric session key. Some answers suggested the use of public and private keys, these were not listed as available in the question text.

Q2) AFC
a) The main problem with answers was that they did not address the fact that reliability means addressing both missing and duplicate data; many answers did not address detecting duplicate data.

b) Some answers indicated that the sender knew by magic how much space was available in the receivers butters rather than via acknowledgements with a window size. They also failed to show the receiver sending an acknowledgement immediately after data was received with the window size of 0. Other mistakes included putting sequence numbers on frames rather than bytes, no delay of one to send a frame across the network, the timing of messages, and the numbering of bytes (the first segment had sequence numbers 96-195, but the second segment was given a sequence number of 197 for its first byte).

c) Answers failed to identify that congestion can be detected via dropped and delayed packets and only mentioned dropped packets. Answers then described flow control rather than congestion control and talked about changing the window size (which is a decision made by the receiver) rather than changing the transmission rate (which is a decision made by the transmitter).

d) Many answers failed to show the working requested by the question text. Some also took a subnetting approach rather than a CIDR approach to the allocation of addresses. Errors included allocation invalid sized blocks of class C address to a network (must be 2^n), not starting an allocated block on a valid address boundary (exactly divisible by the size of the block) and allocating one class C address to more than one network.

Q3) NPF

The question was based on the laboratory exercise but should have been simple to pick up 50-65% of the marks. This question was answered by relatively few students. Answers were very disappointing.

A. Most answers indicated that awareness of IPv6 is limited to having a much larger address space and notation differences in how addresses are often presented in almost human readable form. A few answers were truly worrying in that students showed they did not know IPv4 is based on 32 bit addresses and/or IPv6 on 128 bit addresses.

B. I was shocked how many answers were not aware of how netmasks work. These were used throughout the course.

C. This carried forward the lack of netmask understanding from (b) above.

D. I. Only a few students got this trivial question correct. II. Most answers gave some recognition that tunnelling was required. However, only one answer contained a good overview of how it would work.
E. Most got this correct.

Q4) NPF

I was pleased with the quality of answers to this question which generated a wide spread of marks with the vast majority doing quite well.
A. Good answers.
B. The majority had this yes/no question correct.
C. I was very surprised that almost nobody knew or remembered that almost all increases in data rate happen due to increased bandwidth via increased frequency usage or simply having several wires/fibres! Answers discussed often relatively small effects due to how TCP rate change is managed, router configuration, choosing cables with higher velocity factors for the speed of light etc..
D. This was harder and produced a wide range of answers. Nobody related this to the Shannon-Hartley material which was covered in lectures. Almost nobody knew that complex data modulation schemes such as QAM to carry many bits per symbol are also more difficult to accurately decode. The majority claimed higher data rates cause more errors due to congestion and overflowing buffers causing packets to be dropped. Some answers clearly claim the number of collisions depends on the data rate, at best a weak relationship though used successfully by some candidates. Collisions can of course happen at any data rate due to demand exceeding supply rather than data rate. Broadband, Giga-bit Ethernet etc., would fail badly if they really caused massive congestion, buffer overloads and collisions!
E. Quite a few candidates need to check they understand what reliability is. Answers tended to either give a network view talking about buffer size, no of routes, flow control, acknowledgments or to (what I expected) talk about error detection and correction.
F. Most attempts gave good to very good answers. However, several answered showed they do not know what UDP is!
G. Mostly well answered. Some answers claimed incorrectly that UDP supports reliable multi-media transport. Most candidates realised that UDP with extra layered facilities was needed.