Queen Mary
University of London

First Degrees in Electronic Engineering and Computer Science by Course Units

May 2015 ??:?? - ??:??

ECS404 Computer Systems and Networks Duration: 2 hours 30 minutes

YOU ARE NOT PERMITTED TO READ THE CONTENTS OF THIS QUESTION PAPER UNTIL INSTRUCTED TO DO SO BY AN INVIGILATOR.

Answer ALL Questions

ONLY NON-PROGRAMMABLE CALCULATORS ARE PERMITTED IN THIS EXAMINATION. PLEASE STATE ON YOUR ANSWER BOOK THE NAME AND TYPE OF MACHINE USED.

COMPLETE ALL ROUGH WORKINGS IN THE ANSWER BOOK AND CROSS THROUGH ANY WORK WHICH IS NOT TO BE ASSESSED.

IMPORTANT NOTE:
THE ACADEMIC REGULATIONS STATE THAT POSSESSION OF UNAUTHORISED MATERIAL AT ANY TIME WHEN A STUDENT IS UNDER EXAMINATION CONDITIONS IS AN ASSESSMENT OFFENCE AND CAN LEAD TO EXPULSION FROM QMUL.

PLEASE CHECK NOW TO ENSURE YOU DO NOT HAVE ANY NOTES, MOBILE PHONES OR UNATHORISED ELECTRONIC DEVICES ON YOUR PERSON. IF YOU HAVE ANY THEN PLEASE RAISE YOUR HAND AND GIVE THEM TO AN INVIGILATOR IMMEDIATELY. PLEASE BE AWARE THAT IF YOU ARE FOUND TO HAVE HIDDEN UNAUTHORISED MATERIAL ELSEWHERE, INCLUDING TOILETS AND CLOAKROOMS IT WILL BE TREATED AS BEING FOUND IN YOUR POSSESSION. UNAUTHORISED MATERIAL FOUND ON YOUR MOBILE PHONE OR OTHER ELECTRONIC DEVICE WILL BE CONSIDERED THE SAME AS BEING IN POSSESSION OF PAPER NOTES. MOBILE PHONES CAUSING A DISRUPTION IS ALSO AN ASSESSMENT OFFENCE.

EXAM PAPERS CANNOT BE REMOVED FROM THE EXAM ROOM.

Examiners:
Prof E P Robinson
Dr G H White

Question 1

This question is about the various stages from source code to the execution of a program.

a) What is machine code? What is assembly language? What are the differences between them?

[5 marks]

b) What do the *assembler*, *linker*, and *loader* do when they translate assembly language to machine code and load it into the computer? (Do not worry too much about the precise distinctions between assembler, linker and loader.)

[5 marks]

c) Give an example of a *high level language*. What is a *variable* in such a high level language?

[5 marks]

d) What is the *load-store paradigm*? Explain how this paradigm is used when the CPU calculates with data. Give a *very small* code fragment showing how two variables, `x` and `y`, are added together and the result put in a variable `z`.

[10 marks]

The relevant MIPS instructions are:

```
lw $t1 p              # loads an integer from memory location p
                      # to register $t1
sw $t2 q              # stores an integer from register $t2
                      # to memory location q
add $t1 $t2 $t3       # adds registers $t2 and $t3 together
                      # and puts the result in $t1
```

a) Machine code is the binary representation of instructions as it is found in the CPU of the machine executing them: assembler is a textual description of those instructions, with binary instruction names replaced by symbolic names, RAM addresses replaced by names and with a few other concessions to legibility.

b) The assembler translates the assembly code to binary; the linker replaces symbolic names by their binary counterparts, and the loader loads the program into RAM ready for execution.

c) Java, for example, is a high level language. A variable in such a language is a name for a location in memory, and this name will typically persist so long as the variable is in scope.

d) The load store paradigm prescribes how RAM and registers interact during execution of a program. Computations in the CPU act on data in registers, so before a computation data must be loaded from the RAM into a register or registers; the computation can then take place, and after that the results will be stored back into RAM. A suitable code fragment is:

```
lw $t1 x
lw $t2 y
add $t1 $t1 $t2 sw $t1 z
```

Question 2

This question is about digital representation

Marks will be awarded for working as well as for correct results, and you should therefore include all working in your answers.

a) This part is about conversion between binary and hexadecimal representations.

   i) Explain why it is easier to convert between hexadecimal and binary, or binary and hexadecimal than between hexadecimal and decimal representations.

   ii) Convert the following *hexadecimal* expressions to *binary*:

      1) $3A1$

      2) $ABCDE458$

   iii) Convert the following bit sequences to hexadecimal (sequences arranged in groups of eight bits):

      1) 11110000 11100001 11000011

      2) 10011101 00111100 01111000

[5 marks]

b) This part combines knowledge of 2's complement representations and binary arithmetic. It uses 8-bit representations throughout.

   i) Carry out the following multiplication, taking the numbers as ordinary positive binary:

```
 11011101
 00000011  *
___________


___________
___________
```

   ii) In 8-bit 2's complement, what are the numbers represented by:

      1) 11011101

      2) 00000011

      3) The 8 rightmost bits of your answer for the sum in i) above.

   In each case explain how you get this answer.

   iii) What is the connection between the numbers you have given as answers in 1), 2) and 3).

   iv) State the property of the 2's complement representation that ensures that the relation you have just given holds.

[10 marks]

c) This part is about character sets.

   i) How many bits are used in the ASCII character set? Given that 'A' is character number 65 and 'a' is character number 97, give the bit patterns corresponding to the ASCII representations of 'B' and 'b'.

   ii) How many bits are used in the character set ISO-8859-1? Give the ISO-8859-1 representations of 'C' and 'c' as bit patterns.

   iii) What would be a quick way of checking whether a character was upper or lower case?

[5 marks]

d) This part is about floating point representation.

i) Give an example of a number in (ordinary decimal) scientific notation that uses 8 significant figures, and explain what the term *significant figures* means.

ii) Give an example of two numbers that can both be represented accurately by numbers with 8 significant figures, but whose sum can not be.

iii) Explain why your example is relevant to scientific computation on existing computers, even though they use binary, not decimal.

[5 marks]

a) Students have been given similar exercises though not at such length. Weaker students have learned conversions to and from decimal, rather than direct between binary and hex but have been told explicitly that this is easier. Similar tasks have been set on previous exams (though not using this size of input).

i) Because $16 = 2^4$, a single hex digit corresponds to exactly 4 binary digits. This means it is possible to translate "bit by bit" (or rather "group to group"), only manipulating a part of the string given at each stage. With decimal, this is not possible. (1) for core of this.

ii) 1) $3A1 = 0011\ 1010\ 0001_2$

2) $ABCDE458 = 1010\ 1011\ 1100\ 1101\ 1110\ 0100\ 0101\ 1000_2$

(1) method (1) accuracy

iii) Convert the following bit sequences to hexadecimal (sequences arranged in groups of eight bits):

1) $11110000\ 11100001\ 11000011 = F0\ E1\ C3_{16}$

2) $10011101\ 00111100\ 01111000 = 9D\ 3C\ 78_{16}$

(1) method (1) accuracy

[5 marks]

b) Binary multiplication and conversion to and from 2's complement featured on previous exams, but not put together like this.

i) Carry out the following multiplication, taking the numbers as ordinary positive binary:

$$\begin{array}{rl} 11011101 & \\ 00000011 & * \\ \hline 11011101 & \\ 110111010 & \\ \hline 1111110000 & \text{carries} \\ 1010010111 & \\ \hline \end{array}$$

(3) marks for this.

ii) In 8-bit 2's complement, what are the numbers represented by:

1) 11011101

Begins with 1 so negative. Several methods used by students. One is - flip bits: 00100010, express as decimal: $32 + 2 = 34$, add 1: 35, negate: -35.

2) 00000011

Begins with 0 so positive: use standard binary representation: 3.

3) The 8 rightmost bits of your answer for the sum in i) above. 10010111 begins with 1 so negative. Flip bits: 01101000, express as decimal: $64 + 32 + 8 = 104$, add 1: 105, negate: -105.

(5) marks for this.

iii) What is the connection between the numbers you have given as answers in 1), 2) and 3).
$-35 * 3 = -105$. (1)

iv) State the property of the 2's complement representation that ensures that the relation you have just given holds.
You can add, multiply and subtract signed 2's complement representations using the ordinary unsigned algorithm, ignoring any overflow digits. (1).

[10 marks]

c) Students have been taught about character sets and questions requiring them to give numeric, though not bit, representations of characters appeared on previous exams.

i) How many bits are used in the ASCII character set? Given that 'A' is character number 65 and 'a' is character number 97, give the bit patterns corresponding to the ASCII representations of 'B' and 'b'.
7 bits. 'B' $= 66 = 1000010$. 'b' $= 98 = 1100010$. (2)

ii) How many bits are used in the character set ISO-8859-1? Give the ISO-8859-1 representations of 'C' and 'c' as bit patterns.
8 bits. 'C' $= 67 = 01000011$. 'c' $= 99 = 01100011$. Test mainly to get correct number of bits. (2)

iii) What would be a quick way of checking whether a character was upper or lower case?
Check whether the relevant bit is 0 or 1 (bit 6 counting from right). (1)

[5 marks]

d) This part is about floating point representation.

i) Give an example of a number in (ordinary decimal) scientific notation that uses 8 significant figures, and explain what the term *significant figures* means.
$1.2345678 * 10^0$ Significant figures refers to the number of digits in the decimal expression that is multiplied by a power of 10. (2)

ii) Give an example of two numbers that can both be represented accurately by numbers with 8 significant figures, but whose sum can not be.
$1.0 * 10^0$ and $1.0 * 10^{-8}$. Sum is $1.00000001 * 10^0$ which requires 9 significant figures for accurate representation. (2)

iii) Explain why your example is relevant to scientific computation on existing computers, even though they use binary, not decimal.
Computers use what is essentially a binary version of scientific representation for floating point calculations. So it has similar issues with the consequence that floating point calculations are not perfectly accurate. (1)

[5 marks]

Question 3

This question is about race conditions and deadlock

a) Suppose we have two processors executing at the same time, and both doubling the same variable, `z`. Assume that each of them must first load the value of the variable, then double it by adding it to itself, and then store the doubled value of the variable. There is a potential *race condition*: i.e. the result of this calculation depends on the way that the instructions interleave. Give two different orders of execution which arrive at two different results.

[5 marks]

b) Describe what a *lock* is, and write a code fragment (using operators `lock` and `unlock` which operate on memory locations) which doubles `z`, and which will avoid a race condition

[10 marks]

c) Describe what *deadlock* is, and write a pair of code fragments, one for each processor, which lead to deadlock when executed against each other, and ex- plain how the deadlock occurs. (There is no need to write any code inside the lock-unlock pairs: simply show how the locks are configured)

[10 marks]

a) The following two orders arrive at different results. Note that there is no need to write assembler: the notation I have used is precise enough.

Order 1:

processor 1	processor 2
load z	
double z	
store z	
	load z
	double z
	store z

Order 2:

processor 1	processor 2
load z	
double z	
	load z
	double z
	store z
store z	

???

b) A lock can be used to avoid race conditions: when processor 1 starts its computation, it locks variable z, which prevents any other processor having access to it, and it only releases the lock when its computation is complete. Similarly for processor 2. With locks configured in this way, either processor 1 would complete its calculation before processor 2 starts, or vice versa, and in either case the result would be to multiply z by 4. (I expect that the students will talk in terms of processors rather than in terms of the more abstract concept of processes, which they have not been exposed to.)

c) The following configuration of locks will end in deadlock:

processor 1	processor 2
lock(x)	lock(y)
lock(y)	lock(x)
⋮	⋮
unlock(y)	unlock(x)
unlock(x)	unlock(y)

The processors will end up after the first line when they both have one lock each, and are unable to proceed because the other process has the lock they need to acquire, but cannot release it until it has done the remainder of its task.

Question 4

This question is about computer networks

a) When a webserver is sending a webpage to a browser, the packets typically have four layers corresponding to different protocols being used. List the layers in order, starting at the innermost, giving the name of the layer and the protocol you would expect to see there in this instance.

[8 marks]

b) Draw a simple block-structure diagram of a typical packet including headers and footers, but not detail about the fields in the headers.

[2 marks]

c) Briefly explain the function of the Domain Name Service using the following output from the `dig` command to illustrate your points.

```
[edmundr@bert ~]$ dig mysis.qmul.ac.uk

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.30.rc1.el6_6.1 <<>> mysis.qmul.ac.uk
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51079
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;mysis.qmul.ac.uk.              IN      A

;; ANSWER SECTION:
mysis.qmul.ac.uk.       300     IN      A       138.37.29.35

;; Query time: 1002 msec
;; SERVER: 138.37.0.88#53(138.37.0.88)
;; WHEN: Fri Feb  6 12:07:54 2015
;; MSG SIZE  rcvd: 50

[edmundr@bert ~]$
```

Figure 1: `dig` output

[4 marks]

d) At which packet layers (if any) would you expect to find the following data. Explain your answers in terms of the functionality of the layer and the data.

i) The symbolic name of a webserver.

ii) An IP address.

iii) A port number.

iv) A MAC address.

[8 marks]

e) Describe briefly what the following traceroute output is telling you about the route from `bert.student.eecs.qmul.ac.uk` to `mysis.qmul.ac.uk` (the server for the College's student database).

```
[edmundr@bert ~]$ traceroute mysis.qmul.ac.uk
traceroute to mysis.qmul.ac.uk (138.37.29.35), 30 hops max, 60 byte packets
 1  belt503.eecs.qmul.ac.uk (138.37.37.253)  0.250 ms  0.182 ms  0.185 ms
 2  eecs-sub-backbone-46.core-net.qmul.ac.uk (138.37.2.46)  0.588 ms  0.699 ms  0.718
ms
 3  gooseberry-ebr3.core-net.qmul.ac.uk (138.37.3.207)  0.834 ms  0.821 ms  0.801 ms
 4  138.37.2.22 (138.37.2.22)  1.028 ms  1.018 ms  1.050 ms
[edmundr@bert ~]$
```

Figure 2: `traceroute` output

[3 marks]

a) Essentially bookwork testing basic knowledge of protocol stack.
Layers:
Application: http (hypertext transfer protocol)
Transport: TCP (transmission control protocol)
Network: IPv4 (Internet Protocol)
Link: ethernet of some kind
(1) each layer and (1) each correct protocol.

b) As following from lecture slides:

Ethernet header	IP header	TCP header	Application packet	Ethernet footer

c) The point of the domain name service is to provide a mechanism to translate from a symbolic address (in this instance mysis.qmul.ac.uk) to a digital IP address (in this instance 138.37.29.135). This allows programs to use meaningful (and relatively static) symbolic addresses, rather than the unmemorable and subject to change IP addresses.

d) i) Possibly not there at all. Might be in the application layer, but depends on the particular application being used. http would have it in the headers.

ii) IP addresses would be in the network layer. The job of the network layer is to get the information to the right machine across a number of hops, and the IP address is the addressing mechanism for this.

iii) Port numbers are in the transport layer. The job of the transport layer is to get the application packets to the right programs, and ports give the association of packet to program.

iv) Link layer: MAC addresses are the hardware addresses used on the local links.

e) There's a very short route... only five hops. We start a bert, and then belt503 is the router used by bert, we then go to a gateway from EECS to the college (eecs-sub-backbone) and then a college machine, before reaching an anonymous router 138.37.2.22 and then mysis (138.37.29.135, as in previous question).

End of Paper