Thursday 22 June 2017 Morning Time allowed: 2 hours 30 minutes

Materials
For this paper you must have:
- a calculator.

Instructions
- Use black ink or black ball-point pen.
- Fill in the boxes at the top of this page.
- Answer all questions.
- You must answer the questions in the spaces provided. Do not write outside the box around each page or on blank pages.
- Do all rough work in this book. Cross through any work you do not want to be marked.

Information
- The marks for questions are shown in brackets.
- The maximum mark for this paper is 100.

Advice
- In some questions you are required to indicate your answer by completely shading a lozenge alongside the appropriate answer as shown.
- If you want to change your answer you must cross out your original answer as shown.
- If you wish to return to an answer previously crossed out, ring the answer you now wish to select as shown.
Figure 1 shows some of the internal components of a processor and how the processor is connected to the main memory. The internal connections within the processor are not shown.

Figure 1

<table>
<thead>
<tr>
<th>Processor</th>
<th>Main Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>Address</td>
</tr>
<tr>
<td>MAR</td>
<td>Content</td>
</tr>
<tr>
<td>CIR</td>
<td>0</td>
</tr>
<tr>
<td>MBR</td>
<td>1</td>
</tr>
<tr>
<td>General Purpose Registers</td>
<td>2</td>
</tr>
</tbody>
</table>

Describe how an instruction is fetched from main memory during the **fetch stage** of the fetch-execute cycle.

Your description should cover the use of registers and buses, together with the role of main memory.

[4 marks]
During the decode and execute stages of the fetch-execute cycle the instruction that is being processed is stored in the CIR. Explain why the instruction could not be processed directly from the MBR.

[2 marks]

The computer system shown in Figure 1 uses the von Neumann architecture. The Harvard architecture is an alternative to this.

Explain why the Harvard architecture is sometimes used in preference to the von Neumann architecture.

[2 marks]
Figure 2 shows a message being encrypted using a Caesar cipher.

Figure 2

Wheel settings to use

Example

Plaintext: COMPUTER  
Ciphertext: IUSVAZKX

Decrypt the ciphertext "QGOZRKT" using the Caesar cipher with the settings shown in Figure 2.

[1 mark]

The Vernam cipher is a more sophisticated cipher system that, under certain circumstances, offers perfect security.

State two conditions that must be met for the Vernam cipher to offer perfect security.

[2 marks]

Condition 1


Condition 2


Both the Caesar and Vernam ciphers are symmetric ciphers, whereas a public and private key encryption system is an asymmetric cipher system.

0 2 3

Explain the difference between a symmetric and an asymmetric cipher system.

[1 mark]

Turn over for the next question
In a particular communications system, eight different voltage levels are used to encode the value of groups of bits. Each voltage level encodes the value of one group of bits.

Given that eight different voltage levels are used, how many bits can be in a group that is encoded by a voltage level?

[1 mark]

The baud rate for this system is 500 baud.

What is the system’s bit rate?

[1 mark]

Figure 3 shows three suggested relationships between bandwidth and bit rate.

Figure 3 shows one lozenge to indicate which of the lines, A, B or C in Figure 3, shows the correct relationship between bandwidth and bit rate.

[1 mark]
The system sends the data over a long distance using serial communication.

Explain why serial communication is more appropriate in this instance than parallel communication.

[2 marks]

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

Turn over for the next question
A computer process, X, can only start executing once processes A and B have finished executing and either communication channel C or communication channel D or both are available to use.

The states of processes and communication channels can be read using the following Boolean variables:

- **A** is set to TRUE if process A has completed and FALSE if process A is still running.
- **B** is set to TRUE if process B has completed and FALSE if process B is still running.
- **C** is set to TRUE if communication channel C is available and FALSE if it is not available.
- **D** is set to TRUE if communication channel D is available and FALSE if it is not available.

The Boolean variable **X** should be set to TRUE if the values of the variables **A**, **B**, **C** and **D** indicate that process X can start and to FALSE if they indicate that process X cannot start yet.

In the space below, draw a logic circuit that will represent the logic of the system described above for the inputs **A**, **B**, **C** and **D** and the output **X**.

Write a Boolean expression to represent the logic used to start process X.

\[ X = \text{[Boolean expression]} \]
Using the rules of Boolean algebra, simplify the following Boolean expression.

\[
(A + A \cdot (A + B)) + (B \cdot \overline{C})
\]

You must show your working.

[4 marks]

Answer: ___________________________________________________________________

__________________________________________________________________________

Question 04 continues on the next page
D-type flip-flops can be included in logic circuits.

04.4

Explain the general purpose of a D-type flip-flop. [1 mark]

__________________________

__________________________

04.5

One input to a D-type flip-flop is a data signal.

State what the other input to a D-type flip-flop is and what it is used for. [2 marks]

__________________________

__________________________

__________________________

__________________________
Figure 4 shows the structure of an example machine code instruction, taken from the instruction set of a particular processor.

**Table:**

<table>
<thead>
<tr>
<th>Opcode</th>
<th>Operand(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Basic Machine Operation</td>
</tr>
<tr>
<td>0 1 1 1 1 0</td>
<td>0</td>
</tr>
</tbody>
</table>

State the purpose of the operand part of an instruction **and** explain how the addressing mode is related to this.

[2 marks]

---

Question 05 continues on the next page
Figure 5 shows an assembly language program together with the contents of a section of the main memory of the computer that the program will be executed on. Each main memory location and register can store a 16-bit value.

The assembly language instruction set that has been used to write the program is listed in Table 1 on the next page.

Figure 5

<table>
<thead>
<tr>
<th>Program</th>
<th>Memory Address (in decimal)</th>
<th>Main Memory Contents (in decimal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDR R1, 100</td>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>LSL R2, R1, #2</td>
<td>101</td>
<td>50</td>
</tr>
<tr>
<td>ADD R1, R1, R2</td>
<td>102</td>
<td>80</td>
</tr>
<tr>
<td>LDR R3, 101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMP R3, R1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEQ labela</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOV R4, #0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B labelb</td>
<td></td>
<td></td>
</tr>
<tr>
<td>labela:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOV R4, #1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>labelb:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STR R4, 102</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HALT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Complete the trace table below, in decimal, to show how the values stored in the registers and main memory change as the program in Figure 5 is executed. You may not need to use all of the rows.

<table>
<thead>
<tr>
<th>Register Contents</th>
<th>Main Memory Location Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>R2</td>
</tr>
<tr>
<td>100</td>
<td>101</td>
</tr>
</tbody>
</table>

Explain what the assembly language program in Figure 5 does.

[1 mark]
Table 1 – standard AQA assembly language instruction set

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDR Rd, &lt;memory ref&gt;</td>
<td>Load the value stored in the memory location specified by &lt;memory ref&gt; into register d.</td>
</tr>
<tr>
<td>STR Rd, &lt;memory ref&gt;</td>
<td>Store the value that is in register d into the memory location specified by &lt;memory ref&gt;.</td>
</tr>
<tr>
<td>ADD Rd, Rn, &lt;operand2&gt;</td>
<td>Add the value specified in &lt;operand2&gt; to the value in register n and store the result in register d.</td>
</tr>
<tr>
<td>SUB Rd, Rn, &lt;operand2&gt;</td>
<td>Subtract the value specified by &lt;operand2&gt; from the value in register n and store the result in register d.</td>
</tr>
<tr>
<td>MOV Rd, &lt;operand2&gt;</td>
<td>Copy the value specified by &lt;operand2&gt; into register d.</td>
</tr>
<tr>
<td>CMP Rn, &lt;operand2&gt;</td>
<td>Compare the value stored in register n with the value specified by &lt;operand2&gt;.</td>
</tr>
<tr>
<td>B &lt;label&gt;</td>
<td>Always branch to the instruction at position &lt;label&gt; in the program.</td>
</tr>
<tr>
<td>B&lt;condition&gt; &lt;label&gt;</td>
<td>Branch to the instruction at position &lt;label&gt; if the last comparison met the criterion specified by &lt;condition&gt;. Possible values for &lt;condition&gt; and their meanings are: EQ: equal to NE: not equal to GT: greater than LT: less than</td>
</tr>
<tr>
<td>AND Rd, Rn, &lt;operand2&gt;</td>
<td>Perform a bitwise logical AND operation between the value in register n and the value specified by &lt;operand2&gt; and store the result in register d.</td>
</tr>
<tr>
<td>ORR Rd, Rn, &lt;operand2&gt;</td>
<td>Perform a bitwise logical OR operation between the value in register n and the value specified by &lt;operand2&gt; and store the result in register d.</td>
</tr>
<tr>
<td>EOR Rd, Rn, &lt;operand2&gt;</td>
<td>Perform a bitwise logical XOR (exclusive or) operation between the value in register n and the value specified by &lt;operand2&gt; and store the result in register d.</td>
</tr>
<tr>
<td>MVN Rd, &lt;operand2&gt;</td>
<td>Perform a bitwise logical NOT operation on the value specified by &lt;operand2&gt; and store the result in register d.</td>
</tr>
<tr>
<td>LSL Rd, Rn, &lt;operand2&gt;</td>
<td>Logically shift left the value stored in register n by the number of bits specified by &lt;operand2&gt; and store the result in register d.</td>
</tr>
<tr>
<td>LSR Rd, Rn, &lt;operand2&gt;</td>
<td>Logically shift right the value stored in register n by the number of bits specified by &lt;operand2&gt; and store the result in register d.</td>
</tr>
<tr>
<td>HALT</td>
<td>Stops the execution of the program.</td>
</tr>
</tbody>
</table>

Labels: A label is placed in the code by writing an identifier followed by a colon (:). To refer to a label the identifier of the label is placed after the branch instruction.

Interpretation of <operand2>

<operand2> can be interpreted in two different ways, depending on whether the first character is a # or an R:

- # – use the decimal value specified after the #, eg #25 means use the decimal value 25.
- Rm – use the value stored in register m, eg R6 means use the value stored in register 6.

The available general purpose registers that the programmer can use are numbered 0 to 12.
Discuss the advantages and disadvantages of programming using a high-level language compared to programming using assembly language.

[4 marks]

____________________________________________________________________
____________________________________________________________________
____________________________________________________________________
____________________________________________________________________
____________________________________________________________________
____________________________________________________________________
____________________________________________________________________
____________________________________________________________________
____________________________________________________________________
In a functional programming language a function named `square` and three lists `a`, `b` and `c` are defined as follows.

\[
\text{square } x = x \times x
\]

\[
a = [1, 3, 5] \\
b = [1, 5, 10, 15] \\
c = [9, 7, 2]
\]

What is the list or value that is the result of applying the functions `head(tail(tail b))`?

**[1 mark]**

Calculate the results of making the function calls listed in Table 2 with the lists `a`, `b` and `c` above.

**[3 marks]**

<table>
<thead>
<tr>
<th>Function Call</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>map square a</code></td>
<td></td>
</tr>
<tr>
<td><code>filter (&lt;10) b</code></td>
<td></td>
</tr>
<tr>
<td><code>fold (+) 0 c</code></td>
<td></td>
</tr>
</tbody>
</table>

map is an example of a higher-order function.

Explain what a higher-order function is.

**[1 mark]**
Between 2008 and 2010, a company that was gathering data for an online mapping system, using cars fitted with cameras and WiFi equipment, collected some information that was being transmitted on personal WiFi networks. The company apologised for doing this and an investigation found that a small number of software developers had been responsible for adding this functionality to the mapping system data collection software.

In the context of this example, discuss:

• how it was possible for this data to be collected.
• what steps the owners of the networks could have taken to prevent the data from being collected.
• what legal and ethical issues might have arisen as a result of collecting this data.
• what lessons the company might have learnt from the incident and how their practices might have changed as a result of it.

In your answer you will be assessed on your ability to follow a line of reasoning to produce a coherent, relevant and structured response.

[12 marks]
Two methods of representing and playing music on a computer are sampled sound and MIDI.

Sound is being sampled using a 16-bit sample resolution and a sample rate of 20 000 Hz.

1 Hz is one sample per second.

Calculate the amount of storage space that will be required to store 30 seconds of recorded sound. Express your answer in kilobytes.

You must show your working. [2 marks]

Answer ____________________________
The highest frequency component in the sound that is being sampled is 14 500 Hz. The sample rate of 20 000 Hz is not high enough to enable a faithful reproduction of the original sound from the sample.

Explain why this is the case, justifying your response. [2 marks]

---

MIDI is a system that can be used to enable musical devices to communicate and to represent music on a computer.

Explain how MIDI represents music and the advantages of using MIDI for representing music instead of using sampled sound. [4 marks]
Figure 6 shows the physical topology of a local area network (LAN) and its connection to the Internet. The LAN uses the IPv4 protocol.

State suitable IP addresses for:

- The 'Router and Firewall' port labelled **A** ..............................................................
- The 'Router 2' port labelled **B** ..............................................................

Explain the difference between a physical topology and a logical topology. ..............................................................
Explain the differences between client-server and peer-to-peer networking. [4 marks]

____________________________________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________

____________________________________________________________________________________

Turn over for the next question
A garage services and repairs cars. It uses a relational database to keep track of the jobs that customers have booked for it to carry out. The database includes jobs that have been completed and jobs that are waiting to be done.

The details of the jobs that the garage does, together with the parts that it stocks and uses are stored in the database using the four relations shown in Figure 7.

**Figure 7**

<table>
<thead>
<tr>
<th>Relation</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job</td>
<td>JobID, CarRegNo, JobDate, InGarage, JobDuration</td>
</tr>
<tr>
<td>Car</td>
<td>CarRegNo, Make, Model, OwnerName, OwnerEmail, OwnerTelNo</td>
</tr>
<tr>
<td>Part</td>
<td>PartID, Description, Price, QuantityInStock</td>
</tr>
<tr>
<td>PartUsedForJob</td>
<td>JobID, PartID, QuantityUsed</td>
</tr>
</tbody>
</table>

- Each car has a unique CarRegNo.
- A type of car can be uniquely identified by the combination of its Make and Model. Different Makes may use the same Model name and a particular manufacturer (Make) will produce several different car Models.
- A booking made for a car on a particular date counts as one job, regardless of how many different tasks are completed upon it.
- A job might require the use of any number of parts, including zero.
- Some of the details are stored in the database as soon as a booking is made and others are only added when a job has been completed.

The attribute JobID is the Entity Identifier (Primary Key) of the Job relation.

If the JobID attribute were not included in the Job relation, which other attribute or attributes that are currently in the relation could probably be used as an Entity Identifier (Primary Key) instead?

[1 mark]

It has been suggested that the owner details (OwnerName, OwnerEmail, OwnerTelNo) should not be stored in the Car relation and that a new relation should be created to store owner details separately from car details.

Explain why storing the owner details separately would improve the design of the database.

[2 marks]
On the incomplete Entity-Relationship diagram below show the degree of any three relationships that exist between the entities.

![Diagram]

When an appointment is made for a job, this is represented in the Job relation. At the time of booking, the InGarage attribute is set to False and the JobDuration attribute is set to 0:00. When the car arrives at the garage the value of the InGarage attribute is changed to True. When the job is finished the value of the JobDuration attribute is updated to indicate how long the job took and details of the parts used are recorded in the database.

The Job with JobID 206 has been completed. The job took 1 hour 30 minutes (1:30) and used two of the parts with PartID 12.

Write the SQL commands that are required to record the amount of time that the job took in the database.

![SQL](10.4)

Write the SQL commands that are required to record in the database the fact that two of the parts with PartID 12 were used.

![SQL](10.5)
Figure 7 is repeated below.

**Figure 7 (repeated)**

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job (JobID, CarRegNo, JobDate, InGarage, JobDuration)</td>
<td></td>
</tr>
<tr>
<td>Car (CarRegNo, Make, Model, OwnerName, OwnerEmail, OwnerTelNo)</td>
<td></td>
</tr>
<tr>
<td>Part (PartID, Description, Price, QuantityInStock)</td>
<td></td>
</tr>
<tr>
<td>PartUsedForJob (JobID, PartID, QuantityUsed)</td>
<td></td>
</tr>
</tbody>
</table>

A mechanic needs to produce a list of all of the parts used on the job with JobID 93 for a customer.

This list must include the PartID, Description, Price (each) and QuantityUsed of each part, and no other details. The parts in the list should be ordered by PartID with the parts with the lowest PartIDs nearest to the top of the list.

**10.6** Write an SQL query to produce the list.

[5 marks]
There are restrictions on which parts can be fitted to which cars. For example:

- The driver’s door mirror with PartID 104 can only be fitted to one particular make and model of car.
- The ignition switch with PartID 27 can be fitted to any model of car for one particular make as the maker uses the same ignition switch in all models.
- The tyre with PartID 97 can be fitted to a wide range of cars of different makes and models as it is a standard size.

If the information about which parts could be fitted to which makes and models of cars were represented in the database, it could be used to help a mechanic identify the correct parts to use for a job.

Explain how the database design could be modified to represent which makes and models of car a part can be fitted to.

[3 marks]
A particular computer uses a **normalised** floating point representation with an 8-bit mantissa and a 4-bit exponent, both stored using **two's complement**.

Four bit patterns that are stored in this computer's memory are listed in **Figure 8** and are labelled A, B, C and D. Some of the bit patterns are valid normalised floating point numbers.

**Figure 8**

<table>
<thead>
<tr>
<th></th>
<th>Mantissa</th>
<th>Exponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0 0 0 0 0 0 0 1</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>B</td>
<td>0 1 0 0 0 0 0 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td>C</td>
<td>1 0 0 0 0 0 0 0</td>
<td>0 0 1 0</td>
</tr>
<tr>
<td>D</td>
<td>1 1 0 0 1 1 1 0</td>
<td>1 0 0 0</td>
</tr>
</tbody>
</table>

Shade **one** lozenge to indicate which bit pattern (A–D) in **Figure 8** represents a negative normalised value.

[1 mark]

Shade **one** lozenge to indicate which bit pattern (A–D) in **Figure 8** represents the smallest positive normalised value.

[1 mark]
11.3 The following is a floating point representation of a number:

\[
\begin{array}{cc}
\text{Mantissa} & 1.0110000 \\
\text{Exponent} & 0011
\end{array}
\]

Calculate the decimal equivalent of the number. You \textbf{must} show your working. [2 marks]

\[
\text{Answer: } \frac{\text{Decimal}}{\text{Exponent}}
\]

11.4 Write the normalised floating point representation of the decimal value 58.5 in the boxes below. You \textbf{must} show your working. [3 marks]

\[
\begin{array}{cc}
\text{Mantissa} & . \_ \_ \_ \_ \_ \_ \\
\text{Exponent} & \_ \_ \_ \_ \_ \_ \\
\end{array}
\]

\[
\text{Answer: } \begin{array}{cc}
\text{Mantissa} & 1.0110000 \\
\text{Exponent} & 0011
\end{array}
\]

Question 11 continues on the next page
There can be a loss of precision when a decimal number is stored when using a floating point system.

The closest possible representation of the decimal number 13.8 is shown below.

\[
\begin{array}{c}
\text{Mantissa} \\
0.11011100
\end{array}
\quad
\begin{array}{c}
\text{Exponent} \\
0100
\end{array}
\]

By converting this bit pattern back into denary it can be seen that the actual number stored is 13.75, not 13.8.

1.1.5

Calculate the absolute error that has occurred. [1 mark]

Answer

1.1.6

Calculate the relative error that has occurred. Express your answer as a percentage to two decimal places. [1 mark]

Answer

END OF QUESTIONS