# 7.1 Internal hardware components

## Q1.

**All marks AO1 (understanding)**

| Level | Description | Mark Range |
|---|---|---|
| 4 | A line of reasoning has been followed to produce a coherent, relevant, substantiated and logically structured response. The response covers all three areas indicated in the guidance below and in at least two of these areas there is sufficient detail to show that the student has a good level of understanding. To reach the top of this mark range, a good level of understanding must be shown of all three areas. | 10-12 |
| 3 | A line of reasoning has been followed to produce a coherent, relevant, substantiated and logically structured response which shows a good level of understanding of at least two areas indicated in the guidance below. | 7-9 |
| 2 | A limited attempt has been made to follow a line of reasoning and the response has a mostly logical structure. At least four points have been made. Either a good level of understanding of one area from the guidance has been shown or a limited understanding of two areas. | 4-6 |
| 1 | A few relevant points have been made but there is no evidence that a line of reasoning has been followed. The points may only relate to one or two of the areas from the guidance or may be made in a superficial way with little substantiation. | 1-3 |



**Guidance – Indicative Response**

**For each guidance point, if the student expands on the point to explain in what way the measure will improve performance then this can be considered to be a second point.** For example:

- "Using a processor with more cores" is one point.
- "Using a processor with more cores which will be able to execute multiple instructions simultaneously" is two points.

Note that just "faster" is not enough to count as an expansion point without an explanation of why.

**1. Server Hardware**

Replace the processor with one which has more cores

Replace the processor with one which has more cache memory // increase the

amount of cache memory

Replace the processor with one which runs at a faster clock speed **NE.** faster processor

Use a parallel processor architecture // use more processors which can work in parallel

Use a processor with a bigger word size

Use a processor that makes (better) use of pipelining

Install more RAM // main memory // primary memory

Use RAM // main memory // primary memory with a faster access time

Replace HDDs with SSDs // Replace HDDS with HDDs that can read data at a faster rate

Defragment the HDD

Replace the motherboard with one which has buses which run at a faster clock speed

Replace the motherboard with one which has more lines in the data bus

Use the Harvard architecture

Distribute the processing across multiple servers

**2. Network**

Replace the network cable with cable that has a higher bandwidth // replace copper cable with fibre-optic cable **A.** Ethernet cable for fibre-optic NE. higher bandwidth network

Replace any wireless / WiFi connections with wired ones

Replace the network cards with ones that can transmit data at a higher bitrate

Consider the overall network design eg how the network is divided into subnets **A.** split the network into subnets

Use a star topology (instead of a bus)

Consider using a more efficient protocol for the data across the network

Add additional wireless access points



**3. Database and Software**

Use a more efficient technique for controlling concurrent access to the database // replace record/table locks with serialisation/timestamp ordering/commitment ordering

Replace the database software with software that uses more efficient algorithms for tasks **A.** examples eg replace linear search with binary search

Use the index feature of the database to speed up searching on fields that are commonly used for this purpose

Rewrite the database software in a language that is suitable for concurrent execution // use a functional programming language for the database software

Ensure the software is compiled rather than executed by an interpreter // rewrite the software in assembly language/machine code

Review the conceptual model of the database to see if it contains any inefficiencies such as data redundancy that could be eliminated **A**. normalise the database design

Consider if it would be appropriate to sacrifice normalisation of the conceptual model to improve performance

Use a non-relational database system **A.** examples eg NoSQL

Distribute the data across multiple servers

Try to reduce the amount of other (unrelated) software that might be running on the database server at the same time

Try to reduce the number of database accesses that need to be made simultaneously // run some tasks at quiet times / overnight

Purge / archive data that is no longer necessary / in use

**[12]**

## Q2.

(a) **Marks are for AO1 (knowledge)**

| A | B | Q |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**1 mark:** Table completed correctly;
**1 mark:** AND gate symbol drawn;

**2**

(b) **Marks are for AO2 (apply)**

A.B.(A + B)
A.B.A + A.B.B ; [expansion of brackets]
B.A + A.B ; [use of A.A = A]
A.B ; [use of A + A = A]

**1 mark:** Final answer: A.B;
**Max 2 for working**



**3**

(c) (**Marks are for AO2 (apply)**

X + Y).(X + NOT Y)
XX + X(NOT Y) + XY + Y(NOT Y) ; [expansion of brackets]
X + X(NOT Y) + XY ; [use of X.X = X or use of Y(NOT Y) = 0 ]
X ( 1 + NOT Y + Y ) ; [use of 1 + X = 1]

**1 mark:** Final answer - X;
**Max 2 for working**

**3**

**[8]**

## Q3.

**All marks AO1 (understanding)**

| Correct Name from List |
|---|

| | |
|---|---|
| **B** | Visual display unit; |
| **C** | Processor; |
| **D** | Main memory; |
| **E** | Keyboard; |

**1 mark** per correct answer
**A** If same response used more than once

**[4]**

## Q4.

(a) increase the number of bits that can be transferred <u>at one time</u> ;
**A** increase rate of data transfer;

increases the number of (memory) addresses / addressable locations / /
increase the maximum amount of primary store / memory (possible);

instructions performed more quickly / / instructions executed at faster rate / /
fetch execute cycle will happen faster / /
increased heat may cause malfunctioning of device / / overheating;

**A** calculations / operations / commands for instructions

**3**

(b) (i) a (hardware) device / component that is not part of the CPU;
**NE** processor / computer
a (hardware) device not directly under the control of the processor / CPU;
a device that communicates through an I / O controller;
external hardware / device;
**R** examples alone

**MAX 1**



(ii) to allow exchange of data / instructions / signals between the processor and the peripheral;
**A** communicate
**R** information
**NE** To allow the device to be connected

**1**

(iii) Electronics that interface the controller to the system bus;
Electronics appropriate for sending signals to the device connected to the computer;

**MAX 1**

(iv) Each peripheral operates in a different way;
Not sensible to design a processor to control every possible peripheral;
A new type of peripheral would require the processor to be redesigned;
Peripherals may operate at a different voltage from the processor;
Peripherals will usually operate at a slower rate than the processor (requiring buffering);

**MAX 2**

**[8]**

## Q5.

(a) AND;
NOR;
XOR; **A** EXOR // EOR // NEQ // exclusive OR;

**3**

(b) (i) B

**1**

(ii) B

**1**

(iii) 0;;

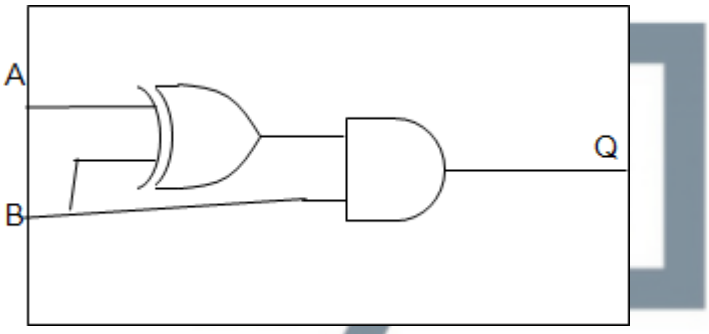
Award 1 mark if De Morgan's has been applied once correctly but candidate does not end up simplifying to 0

Example: $\overline{B + (\bar{A} + \bar{B})}$

Example: $\bar{B} . (A . B)$

**2**

(c)




Inputs A and B connected to an XOR gate;

Input from B and output of XOR gate connected to an AND gate with output going to Q;



**2**

**[9]**

## Q6.

(a) A set of / group of / parallel wires / lines;

*Wires needs to be qualified with set / group*

that are used to connect together components (inside the computer) // connect different parts of the CPU

in order to pass signals between them;

**R** a wire
**A** connect different parts of the computer
**NE** data

**Max 2**

(b) Instructions;

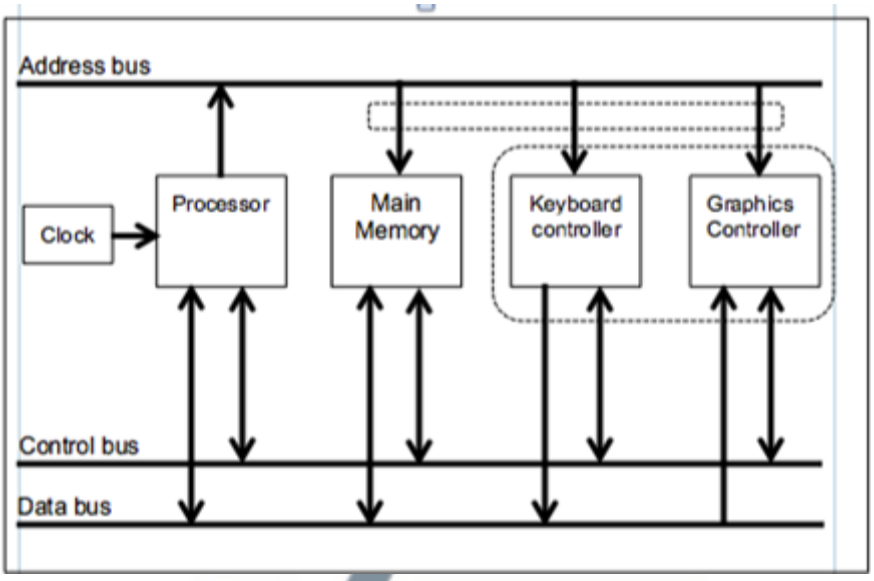
**A** Commands / machine-code
**R** signals

***Examples of a control signal (Max 1):***
**NE** an event that details when an interrupt would be caused

Clock / timing; reset; interrupt ACK; interrupt request; bus grant; bus request; status; I / O write; I / O read; memory read; memory write; transfer ACK
**A** interrupt **A** transfer request
**A** read / write
**NE** load / store
**NE** clock speed

**2**

(c)




1 mark – one of processor, keyboard controller or graphics controller identified correctly
2 marks – all three correctly identified



Address bus connects the 4 components;
Arrow from processor to the address bus;
Arrows from address bus to the three other components;

*Mark this on where the candidate has put the components.*

**5**

**[9]**

## Q7.

(a) Address (bus);

**1**

(b) 1;
**R** 33

**1**

(c) A – Visual display unit; **A** VDU
B – Processor; **R** CPU
C – (Main) memory;
D – Keyboard;

4

**[6]**

## Q8.

| (a) | Number | Component Name |
|---|---|---|
| | 1 | Memory Address Register |
| | 2 | Address Bus |
| | 3 | Memory Data / Buffer Register |
| | 4 | Data Bus |

**4**

(b) The instruction is held in the CIR;
**A** IR
The control unit / instruction decoder decodes the instruction;
The opcode identifies the type of instruction it is;
Relevant part of CPU / processor executes instruction;
**A** ALU
Further memory fetches / saves carried out if required;
Result of computation stored in accumulator / register / written to main memory;
Status register updated;
If jump / branch instruction, PC is updated;
**A** SCR

**Max 3**

(c) Can be displayed in less space;
**R** takes up less space **NE**
Easier to remember / learn / read / understand;
Less error prone;

**Max 1**



(d) (i) Assembler;

**1**

(ii) HLLs are problem oriented;
HLL programs are portable // machine / platform independent ;
English like **keywords / commands/ syntax / code**;
**R** closer to English
Less code required // less tedious to program //
one to many mapping of HLL statements to machine code commands;
Quicker/easier to understand / write / debug /learn / maintain code;
**R** just quicker/easier
HLLs offer extra features e.g. data types / structures // structured statements // local variables // parameters // named variables/constants;
**R** procedures / modular
**A** example of a data structure
**NE** "extra features" without example
Speed of execution not crucial for most tasks so faster execution of assembly language not required;
Most computer systems have a lot of (main) memory / RAM so compact object code not essential;

**A** converse points for Assembly Language

**3**

**[12]**

## Q9.

(a) Program Counter;
**A** Sequence Control Register
**R** Next Instruction Register
Current Instruction Register;
**A** Instruction Register
Memory Buffer Register;
**A** Memory Data Register
Memory Address Register;

**Max 2**

(b) Address in MAR/address to fetch instruction from, sent down Address Bus to Main Memory;
**R** address in PC (program counter)
Contents of address accessed in Main Memory;
**A** by implication if
contents of address location referred to during data transfer
Contents of address location//instruction//data passed down Data Bus into MBR/to processor;
**A** MDR instead of MBR
**A** RAM for Main Memory

**Max 2**

(c) Order of execution unimportant/one step does not rely on prior completion of the other;
Steps carried out by different (hardware) devices/components;
**A** operations are independent
**A** operations use different registers
**R** using different buses

**Max 1**

**[5]**



## Q10.

| **Internal Components** | |
|---|---|
| Data Bus | 10 |
| Address Bus | 9 |
| Control Bus | NA |
| *VDU Controller* | 8 |
| Disk Controller | NA |
| *Keyboard Controller* | 7 |
| Main Memory | 5 |

| **Peripherals** | |
|---|---|
| *Keyboard* | 2 |
| *Visual Display Unit* | 3 |
| Secondary Storage | NA |

| | |
|---|---|
| Processor | 4 |

*1 mark for each correct answer (10,9,5,4)*
*1 mark for correct pair (8,3)*
*1 mark for correct pair (7,2)*
*Mark diagram if answers written on it instead of in tables. Answers in tables over-ride answers on diagram.*

**[6]**

## Q11.

(a) **Step 1**: MAR ←[PC] / Contents of program counter transferred to MAR;

**Step 2b:** MBR ← $[Memory]_{addressed}$ / Contents of addressed memory location loaded into MBR; (must have concept of data coming from address in memory, not just going into MBR)

**Step 4:** Decode instruction;
**A** Contents of CIR decoded
**R** Data for instruction
**R** CIR decoded, CIR decodes instruction

*1 mark for each correct step*

For PC accept Program Counter / SCR / Sequence Control Register
For MAR accept Memory Address Register
For MBR accept Memory Buffer Register / MDR / Memory Data Register
For CIR accept Current Instruction Register / IR / Instruction Register
**A** Other means of indicating correct transfer e.g. [PC] → MAR or MAR:=PC
**A** Missing square brackets or alternative types of brackets
**A** Answers that miss out reference to "contents of"
**A** [Memory] for $[Memory]_{addressed}$

**3**

(b) (i) Increases the number of bits (**A** amount of data) that can be transferred at one time // increase rate of data transfer;

**1**

(ii) Increases the number of memory addresses / /Increase the maximum amount of primary store/memory (possible);

**1**

(iii) Instructions performed more quickly // Instructions executed at faster rate;
**A** Calculations for instructions (this time only)
**A** Operations for instructions
**NE** Speeds the computer up
**R** Processes, tasks for instructions

**1**

**[6]**

## Q12.

(a) **Processor/CPU;**
Explanation faster execution of (program) instructions / the fetch-execute cycle is faster;

**R** more 'calculations per second'
Simultaneous processes possible / duel/quad – core processor;

**Max 2**

**Additional processor**;
Processing is shared between two processors;

**Graphics Card**;
Explanation – increasing the speed at which images are rendered;

**(main) memory / RAM**;
Explanation – reduces main memory to disc transfers;
Fit memory which has a faster read/write speed;

**R** Clock
**A** Explanation – increasing the clock speed/over-clocking;
**R** Cache
**A** Explanation – program instructions are fetched faster from cache than main memory;

(b) **Secondary storage/memory/disc store** // **(external) hard disk**;
**A** HDD/ Hard drive
Explanation – the storage space/capacity is increased;
**R:** 'bigger hard drive' or similar

**2**

(c) **Hub device / USB ports**;
**Card with additional serial /parallel ports / PCMCIA / USB ports;**
**R** Card with additional I/O ports
Explanation – will allow/support the simultaneous connection of several devices;

**Max 2**

**[6]**

## Q13.

(a) (i) 1 GB ;

**1**

(ii) 300 GB ;

**1**

(b) Control (bus) ;

**1**

(c) Data bus has to transport data values to and from various devices /internal components ;
Only the processor assigns address values to the different devices ;

**Max 2**

(d) Logical // read // write // jump/branch // input // output // data transfer ;
**A** Boolean

**1**

(e) Program instructions are transferred from backing store to main memory ;
Program consists of a sequence of instructions ;
Program is stored in main memory ;
and can be replaced by another program at any time ;
Instructions are fetched (in sequence) ;

Decoded ;
and then executed ;

**Max 3**

**[9]**

## Q14.

(a) Halve the time to perform an operation; **A** Operations performed more quickly;

**1**

(b) Increase the number of bits transferred at any one time from 16 to 32// Double the number of bits transferred at any one time;

**1**

(c) Increase the number of memory addresses; from $2^{24}$ to $2^{32}$;

**2**

**[4]**

## Q15.

(a) Electrical/electronic/physical components/parts of computer/system;
**A** any example e.g. motherboard
**R** peripherals(b)

**1**

(b)
- C - **processor / CPU;**
  B - faster execution of program instructions / programs; simultaneous processes possible;
  **A** computer runs faster
- C - **(main) memory / RAM**;
  B - more programs resident in memory; reduces main memory to disc data transfers / programs execute faster;
  **R** stores more data
- C - **secondary/disc store / hard disc;**
  B - more programs/data can be permanently stored / available; faster access/loading speed;
- C - **motherboard**;
  B - allows for faster execution of programs / connection of new I/O ports (e.g. USB / Firewire);
- C - **(3D) graphics/video card**;
  B - display of high resolution /3-D graphics / maximise the benefits possible from some software / better quality images // dedicated processor;
- C - **Sound card;**
  B - better quality sound / surround sound
- C - **Modem**;
  B - External communication /e.g. connect to the Internet
- C - **Network card / NIC**;
  B - communication with other PCs // provides some external communication/connection;

- C - **CD drive**;
  B - higher read/write speed

- C - **DVD drive**;
  B - increased storage /e.g. higher quality media;
  **R.** Router / Cache

**Max 4**

(c) A Sound card / Graphics card / Modem / Network card / Main memory if not given in (b)

- C- **new I/O port**
  **A** example e.g. TV input/USB;
  B - connection to devices which were not previously possible / connection to an additional device e.g. a second parallel printer;
  **R** Motherboard / CPU/ hard disc / CD drive / **DVD** drive

  Sound card
  B - to output audio on the PC (for the first time) - accept as a B. for (c) only

**Max 2**

**[7]**

## Q16.

(a) (i) **Do not mark this part**

(ii)

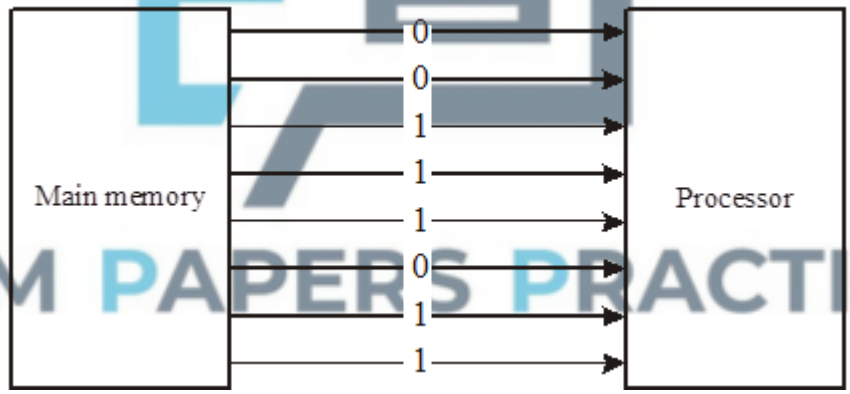



Diagram has 8 lines;
Diagram has the correct 1 bit per line; A**.** pulses to indicate 0/1's

**2**

**A** either 'top to bottom' or 'bottom to top' labelling of the bits
**I.** the direction of any arrows

(b) Interpretation:

- program instruction/command;
- character / ASCII code / 7 data bits + parity bit;
- integer / 59 / number;
- real / number;
- byte/pixel from a graphics file; R. 'part of
- byte/sample from a sound file; R. 'part of
- an address;

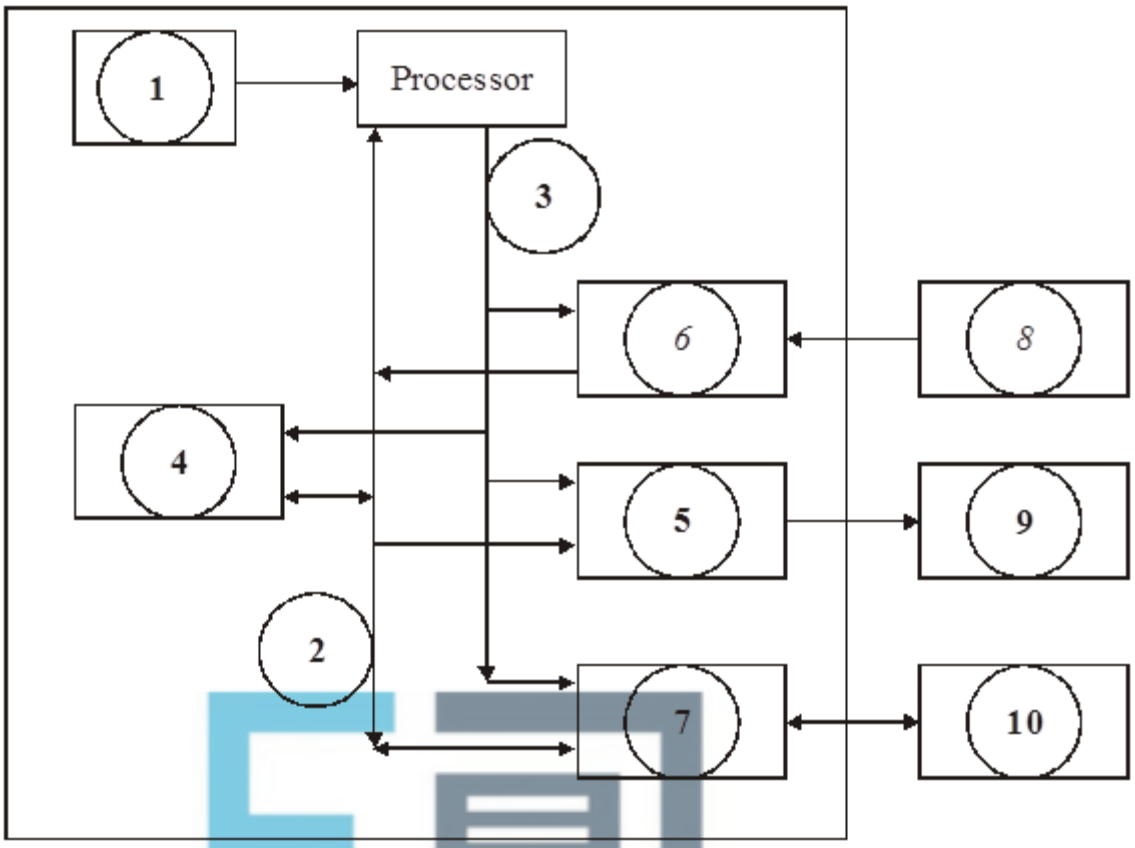
**R** BCD digits

**Max 3**

**[5]**

## Q17.

(a)




*1 mark each for 1, 2, 3, 4.*
*1 mark for both 5 & 9*
*1 mark for both 7 & 10*

**6**

(b) Machine code instructions/program stored in main memory; **A** RAM/IAS
Can be replaced by another program at any time;
**A** Fetched and executed; (concept)

**1**

**[7]**



## Q18.

(a) 1 Clock;
2 Processor;
3 read only memory;
**A** ROM;
4 random access memory;
**A** RAM;
5 data bus;
6 address bus

**6**

(b) Address Bus;
**A** 6 if correct in (a)

**1**

(c) Data Bus;
**A** 5 if correct in (a)

1

**[8]**

## Q19.

(a) (i) Read Only Memory;

1

(ii) Random Access Memory;

1

(b) (i) Disk Controller;

1

(ii) Network Interface Card//Network Adapter;
**A** Network Card

1

(c) Address Bus;
Control Bus;
Data Bus;

3

(d) Program stored in main memory;
**A** RAM/IAS
**R** ROM
Instructions fetched and executed by processor *(concept)*;
**A** CPU
Can be replaced by another program;
**R** cache

2

(e) (i) The number of 1s (including the parity bit) comes to an even number;

1

(ii) Used to check for errors when data is read / transferred;
Parity bit regenerated / recalculated;
Compared with parity bit;

*Any 2*

2

**[12]**

## Q20.

(a) Data bus;
carries data to/from processor / memory / devices /components;
Address bus;
carries addresses / identifies locations;
Control bus;
carries control signals / controls devices;
**A** by example

*Max 1 mark for carries Data / carries addresses / carries control signals*

6

(b) Network adapter / network card;
**A** named example e.g ethernet card generate / understand signals / data (that conform to the LAN protocol) /

Allows (successful) communication / Provides a unique network address;
**R** connect

**2**

(c) Faster transmission;

**1**

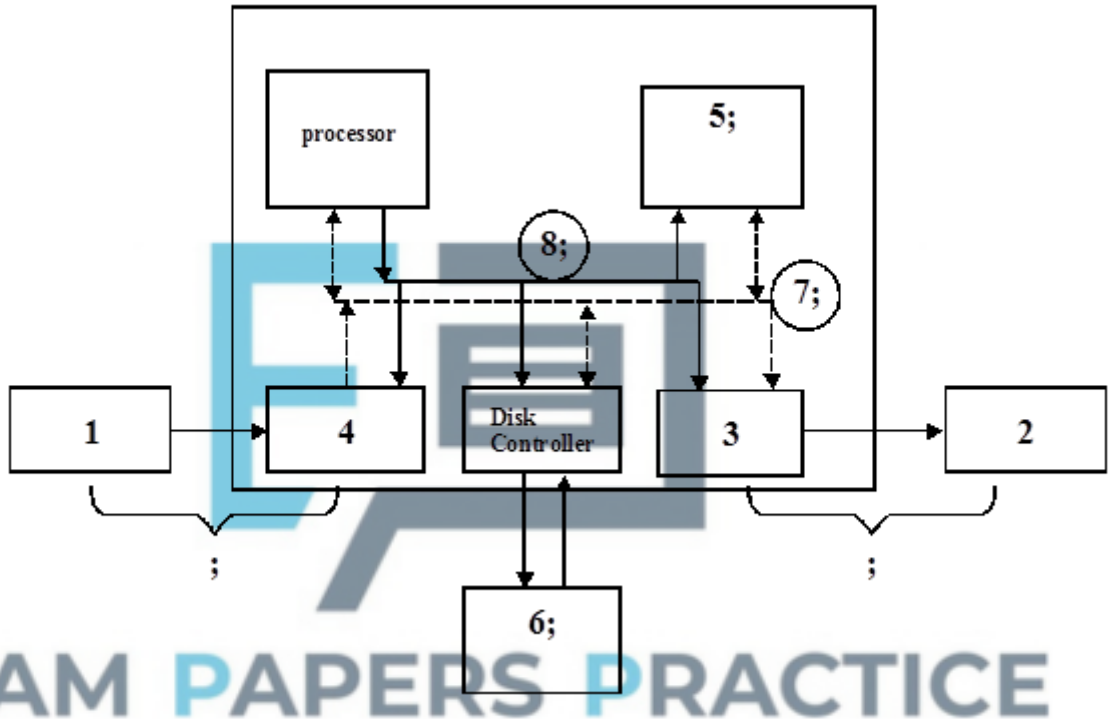
(d) Data transmitted longer distance than is possible with parallel / less expensive to cable;
**R** cheaper

**1**

**[10]**

## Q21.

(a)






*Allow names instead of numbers*

**6**

(b) Machine code instructions/program stored in main memory; **A** RAM/IAS;
Fetched and executed; *(concept)*
Can be replaced by another program any time;
**R** cache
**R** ROM

**2**

**[8]**

## Q22.

(a) 1 – main memory;
2 – processor; **A** CPU;
3 – I/O port;
4 – address bus;
5 – control bus;
**R** anything else

5

(b) (i) (Processor) executes instructions; **R** data **R** programs **BoD** 'executes data and instructions';
(main memory) stores program/data currently in use;
A temporary storage of data/programs; **R** information **R** application
(secondary storage) holds programs/data/files for long-term/ non-volatile storage;
**R** application **I** virtual memory
**A** permanent storage of data/programs **R** information **R** backup

3

(ii) Clock/timing; reset; interrupt ACK; interrupt request; bus grant;
Bus request;
Status; I/O write; I/O read; Memory read; memory write;
Transfer ACK;
**A** interrupt;
**A** transfer request;
**A** examples *read/write on its own not enough*

**Max 2**

(iii) Instruction(s); address(es);

**Max 1**

**[11]**

## Q23.

(a) 1 – processor;
2 – main memory;
3 – keyboard controller;
4 – VDU controller;
5 – data bus;
6 – address bus;

6

(b) 255 / 28-1;

1



(c) (i) Main memory / primary memory / RAM /
Immediate Access Store / IAS ;
**A** ROM ;
**A** cache memory;
**R** registers;
**R** processor/CPU
**R** hard drive
**R** memory

1

(ii) Main memory / primary memory / RAM /
Immediate Access Store / IAS;
**A** cache memory;
**A** registers;
**R** processor/ CPU
**R** hard drive
**R** memory

1

**[9]**

## Q24.

(a) Processor would have to be re-designed; (1)
Every time a new type of device was connected. (1)
**OR**
Voltages / signals required for correct operation of device; (1)
Different from voltages / signals used by processor.(1)
**OR**
So as not to slow the processor down; (1)
To cut down on the number of required ports;(1)
1 device controller can control more than one device of the same type;(1)

**2**

(b) *nb 'controller' or 'card' needed*
Floppy disc
Hard disc / IDE
Any serial device - mouse, printer, modem, Joystick
Any parallel device - printer, CD-ROM, CD-R, DVD, tape unit, zip drive, scanner.
SCSI - Zip drive, tape unit, CD-ROM, CD-R, DVD
USB
I/O controller
Sound card / MIDI interface card / graphics card
Network controller card
*Any one*
**R** keyboard, VDU

**1**

(c) *Any 2 points*
Address only goes (from the processor) to device controllers / main memory;
Regardless of whether the data is to be read from or written to that location;
No feedback / acknowledgement;
Any data transfer goes on data bus;

**2**

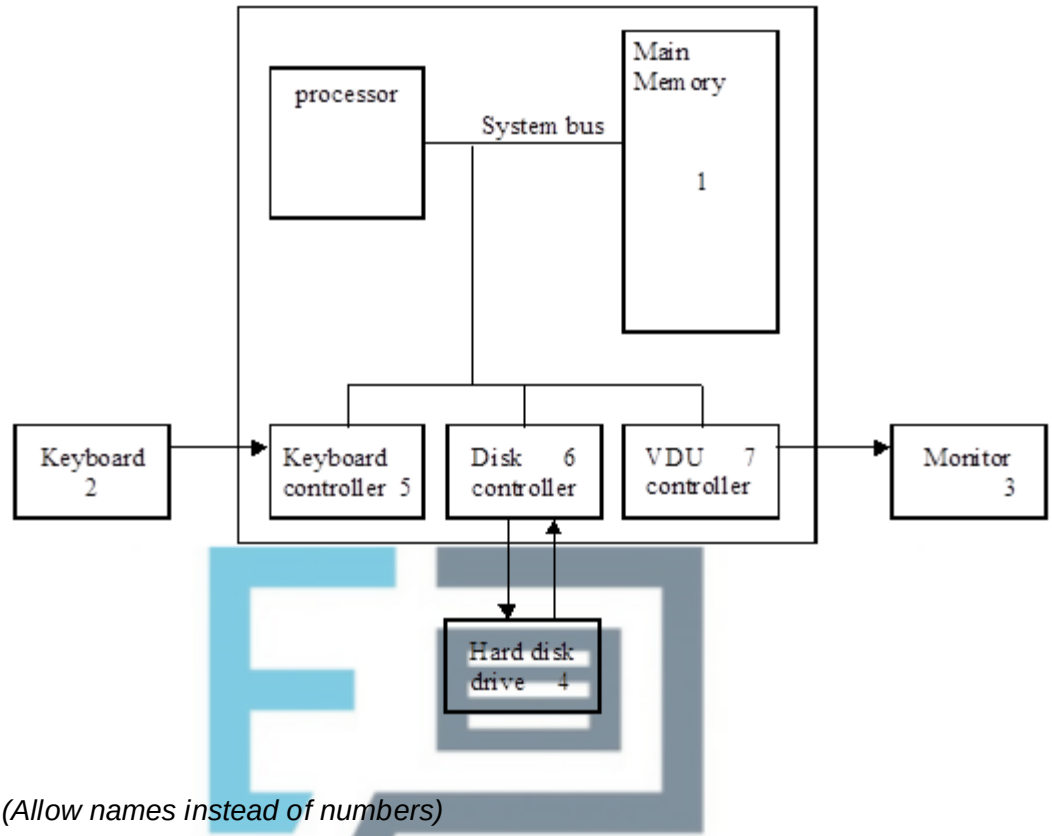
(d) *Any 2 × two each (1 for name, 1 for description)*
Memory Write: causes data on the data bus to be written to the addressed location
Memory Read: causes data from the addressed location to be placed on the data bus / MBR / MDR
I/O Write: causes data on the data bus to be output to the addressed I/O port
I/O Read: causes data from the addressed I/O port to be placed on the data bus
Transfer ACK: indicates that data have been accepted from or placed on the data bus
Bus Request: indicates that a component needs to gain control of the system bus
Bus Grant: indicates that a requesting component has been granted control of the system bus
Interrupt request: indicates that an interrupt is pending / transmission error
Interrupt ACK: acknowledges that a pending interrupt has been recognised
Clock / Timing: used to synchronise operations
Reset: initialises all components.
**A** combined Memory, I/O, Bus, Interrupt lines with suitable explanation

**4**

**[9]**

## Q25.

(a) Correctly placed labels: main memory (1);
Keyboard & keyboard controller (2,5);
Disk controller & HD drive (6,4);
Monitor & monitor controller (7,3);




*(Allow names instead of numbers)*

**4**

(b) Machine code instructions/program stored in main memory/RAM/IAS; fetched and executed; *(concept)*
Can be replaced by another program any time;
**R** cache

**Max 2**

(c) (i) Bits are sent one after another / bits sent one at a time / bits sent singly / along a single wire/line;
**R** data / character

**1**

(ii) Bits transferred simultaneously/concurrently / bits sent down many wires at the same time;
**A** diagram;
**R** data / character

**1**

(d) (i) Between devices in close proximity / communication within computer / communication over short distances;

**1**

(ii) Distance: parallel transmission only operates over short distances;
speed: parallel transfer faster than serial;

**1**

(e) Start bit marks beginning of character to be transmitted / alerts/synchronises receiving device (1)

Stop bit(s) marks end of character to be transmitted / gives time for receiving device to recover; frames the character (1)

**2**

**[12]**

## Q26.

(a) (i) Random Access Memory

(ii) Read Only Memory

*In each case 1 mark for name + 1 for description*

**4**

(b) Bootstrap Program
System Constants

*Any 1*

**1**

(c) User Data
Application Software
System Software
System Variables
Buffers
Disk Cache

*Any 3*

**3**

**[8]**

## Q1.

A very good range of responses was received to this question, with approximately half of students achieving five or more marks. Most students addressed all three aspects of the question (hardware, network, database and software). Students tended to make more points about how the hardware could be improved than about the other two areas. This was acceptable but students needed to have covered all three areas to achieve a mark of ten or above.

Some students wrote too vaguely to achieve marks, for example by writing that a “faster processor” would improve performance, without referencing a factor such as the clock speed that would make the processor faster. Other mistakes included believing that the question required students to contrast thin-client and thick-client and that the system was web based.

A small number of students wrote about issues which might be causing the system to perform poorly instead of explaining how the performance of the system could be improved. Such responses were not worthy of a mark.

## Q4.

(a) The content of this part has appeared in a similar format in a previous paper and a third of students achieved 2 or 3 marks for this question part. Candidates who secured marks could clearly distinguish what performance changes would occur in a way that was precise and tied in to how a computer works. Answers such as 'the computer would be faster' did not gain a mark when thinking about increasing the clock speed as this does not contain the depth that is expected from an AS-level candidate. Discussing how more instructions would be executed per unit time did secure the mark. Candidates also seemed confused over the address bus with answers along the lines of being able to get more memory addresses per unit time rather than discussing that an increase in the width of the address bus would lead to the CPU being able to access more addressable memory locations.



(b) This part was based around I / O controllers, which is an area of the specification that has not been asked about previously. It was obvious that candidates did struggle with this part and this could be due to it being a topic that has not actually been taught or a topic that has only been covered very briefly.

(i) A lot of candidates could describe what a peripheral was and secured the mark for this section. Candidates who talked about a component outside of the processor did not secure the mark as this could also include main memory which is not considered to be a peripheral.

(ii) A group of candidates could discuss the role of the I / O port in terms of allowing data to be transferred from the peripheral to the CPU. Others talked about allowing communication between peripheral and CPU and this also secured the mark.

(iii) As this topic proved to be hard not many candidates could describe another part of the I / O controller. Those that could generally talked about electronics allowing the connection to the system bus or considered the electronics necessary to connect the peripheral to an actual physical port.

(iv) In this part was a question that candidates attempted and many successfully

secured marks. A popular reason, that was allowed for a mark, was the idea that it might slow down the processor. It was pleasing to see stronger candidates talk about the idea of a peripheral operating at a slower speed than the processor and some talked about the need for a buffer. The idea of a peripheral working at a different voltage to the processor was another popular reason that secured a mark.

## Q5.

The majority of students performed well across this question but it was the Boolean simplification parts that caught out some students.

Part (a) had over half of all students gaining full marks for identifying the logic gates for each truth table correctly. The NOR gate proved to be the most challenging truth table to decipher. The three simplification parts appeared to get progressively harder for students working through the paper. It is to be noted again that students do appear to understand that a law of De Morgan needs to be used but they apply it across the whole expression rather than carefully separating it into a left and right part with one operation in the middle.

Part (c) had over 80% of students securing full marks and was answered well. Mistakes came only from students placing the gates in the wrong order or drawing gates incorrectly.

## Q6.

The majority of students scored at least one mark when explaining what was meant by the term bus. This was generally achieved by discussing how a bus connects components together inside a computer. It was not very common to see a good discussion about the use of parallel wires with many students just stating ‘a wire’, however students were still able to pick up full marks by providing other valid points. Weaker students talked about information being carried by a bus rather than signals and gave vague statements about carrying this ‘back and forth’.

Part (b) proved to be harder for students to secure marks on and they struggled to provide a clear example of a control signal. Some students described situations that might cause an interrupt to be generated, for example, the pressing of a key on a keyboard, but usually these events were not themselves control signals. The common correct answers included a memory read or memory write signal with the more able students explaining this in further detail.



The diagram in (c) was answered well with the majority of students securing 3 or more marks. The directionality of the arrows for the address bus was the main cause for students to lose marks and it was surprising to see answers where no attempt was made to connect up the address bus. Students must be reminded to read questions carefully.

Of the students who did connect up the address bus a few did not realise that the keyboard and graphics controller would have a connection to the address bus.

## Q7.

Part (a) was answered correctly by the majority of candidates, but some failed to secure the mark. Most of those who answered incorrectly gave the response ‘control bus’. The only unidirectional bus is the address bus.

Part (b) was poorly answered by the majority of candidates. Only a few candidates identified that one extra line would be needed to go from being able to access 4 GB of main memory to being able to access 8 GB. Candidates gave many varied answers with the most common being 32, 33 and 64. Whilst some candidates may have misread the

question, it is clear that this topic was very poorly understood.

Part (c) was very well answered with the majority of candidates scoring highly. The most common mistake was to swap around processor and main memory; noting the direction of the address bus would have avoided this.

A few candidates used the answers 'address bus' and 'data bus' and it is clear that the bus system still causes confusion for a minority.

## Q8.

This question covered another aspect of computer hardware, the fetch–execute cycle, and why programs are written in high level languages. The table was correctly completed in the majority of cases with the labelled parts of the processor. However some answers simply gave the acronyms rather than the full names of the registers, which the question had clearly asked for. Questions about the 'decode and execute' parts of the cycle have not been asked before this showed in the answers with many candidates describing the fetch part of the cycle and not what was asked. The part of the question concerning why programmers prefer instructions in hexadecimal compared to binary was often answered by saying it takes less storage space which clearly it does not. The answer to the question about the program translator was almost universally well known. When answering the final part, worth three marks, about why programs re written in a high level language, candidates often gave only two reasons and automatically failed to gain one mark. Answers stating that it is, 'like English,' were simply not enough and were marked accordingly; candidates needed to add to this to gain a mark.

## Q9.

For part (a) the vast majority of candidates were able to name two registers that were involved in the fetch part of the fetch–execute cycle, and overall a wide range of the possible answers were given. A minority of candidates gave acronyms such as MAR and so failed to gain credit. Some candidates clearly knew the correct acronyms and tried unsuccessfully to invent or recall a corresponding full name, e.g. "Memory Bus Register", "Memory Access Register".



For part (b) candidates often attempted to describe what was happening in steps 1 and 2b without explaining the role of the two buses and main memory, which is what had been asked for. Very often candidates confused the roles of the data and address buses. There were many references to bidirectional data buses and unidirectional address buses but this did not answer the question that had been asked. Candidates often stated that the memory was moved rather than the "contents of" a specific location or its equivalent description. Also fairly common as a wrong answer was the fact that the address bus carried data from place to place.

Responses to part (c) were better than those to part (b). Nevertheless many candidates did not realise that the two processes are independent of each other. A very common wrong answer was that the buses are bi-directional and so allow two things to happen in opposite directions simultaneously. The idea that the steps 2a and 2b occur in different components was better known but often very vaguely stated. Candidates sometimes talked themselves out of a mark by stating the processes happen in different registers and different buses.

## Q10.

This question was very well answered, with many candidates scoring full marks. The key to answering this type of question successfully is to identify the address bus (as it is unidirectional) and the processor (as it the only device that outputs onto the address bus).

## Q11.

The best answers to this question were given by candidates who had learned the register transfer notation for the fetch-execute cycle and could therefore describe each step concisely and unambiguously. Some good answers were written as prose but candidates who took this approach often lost marks through either lack of understanding or inaccurate expression.

The most common misconceptions were in steps 2b and 4. During step 2b, the contents of the memory address stored in the MAR are fetched from the main memory into the MBR. The contents of the MAR are not simply transferred into the MBR. During step 4 it is the contents of the CIR that are decoded, not the CIR itself. As in January, many candidates attributed the CIR with the ability to do the decoding instead of realising that the CIR simply holds the instruction whilst it is decoded.

It was pleasing to see that many candidates understood that increasing the width of the data bus would enable more data to be transferred at one time. In contrast, only a small number of candidates understood that increasing the width of the address bus would increase the amount of addressable memory. Many mistakenly believed that it would allow multiple addresses to be sent simultaneously. Candidates appreciated that increasing the clock speed would speed the computer up, but many failed to explain that this would be because instructions could be executed more quickly so failed to gain credit.

## Q12.

In part (a) the most popular answers were ‘processor’ and ‘RAM’ but few candidates were able to give a convincing answer for the second mark, often merely re-stating the words in the question stem that the processor would ‘increase the speed of execution of the program’. A good explanation for RAM was rare; candidates clearly have the knowledge that more RAM should result in the faster execution of a program but not the reason behind it.

The most popular answer for (b) was some form of hard drive with this time candidates usually able to get the second mark for a simple statement that it will, 'increase the storage capacity,' but note 'bigger disk' was not considered sufficient (a good example where clear quality of expression gets the mark).



For (b) 2 marks were rarely scored. Candidates who had used a USB on their computer would have identified with this question straight away, but an answer stating that the devices can be then ‘simultaneously’ connected was often missing.

Worryingly, as the content would seem to be at the heart of the subject at this level, there were a noticeable number of scripts which did not attempt some or all of this question. The distribution of candidate marks indicated that this question proved to be a good discriminator of candidates’ ability.

## Q13.

(a) Generally well answered.

(b) Generally well answered.

(c) Few candidates scored the full two marks. The idea that the data bus is used to transport data values typically from the processor to main memory, but also from the memory to the processor (hence requiring the bus to be bidirectional) - the address bus only ever transports addresses from the processor to the various devices; this was the level of answer expected.

(d) The specification says “arithmetic and logical operations”. Credit was given to candidates who clearly had some practical experience of assembly language and so quoted particular types of operations from the instruction set. We stress that although practical experience of assembly language is not a CPT1 requirement candidates do need to understand that a processor performs basic machine operations.

(e) What was asked for was an understanding of the stored program concept and, despite this appearing on several previous CPT1 papers, answers given continually fail to describe this fundamental concept of how a digital computer works.

## Q14.

This was a very straightforward question but the responses were often disappointing. Parts (a) and (b) were generally done quite well but fewer candidates were able give a satisfactory response to part (c). Of those that did answer part (c) many failed to gain the second mark. This was another example of poor examination technique. Candidates should be encouraged to note the number of marks for a question and take this into account when deciding on their response.

## Q15.

(a) Most candidates scored the one mark.

(b) Although not examined on previous papers, candidates scored highly on this question no doubt able in many cases to draw on their own experiences.

Some candidates missed out on full marks with a weak explanation of the benefit to be gained – e.g. ‘more storage’ on its own is not a benefit for changing the hard drive.

A worrying misconception of a few students was that it is possible to change in isolation either the data bus or address bus.

(c) Again well answered with the majority of students able to suggest an additional component. Candidates who did not appreciate the meaning of PCB then wrongly suggested components such as DVD drives or an additional hard drive. More worrying were the candidates who suggested the additional component could be a motherboard or the processor.



## Q16.

(a) Due to an error in the question paper for part (i), examiners were instructed not to mark this part question. The total mark for the paper was therefore reduced to 64. Despite a different style of question for part (ii), the vast majority of candidates were clear as to what was required. Common errors were a diagram which showed 9 lines instead of 8 or the lines drawn as ‘tubes’. Some candidates drew pulses on the lines instead of labelling each one with a 1 or 0 and still scored both marks.

(b) This was surprisingly badly answered with candidates either not clear as to what was being asked for, or not understanding the phrase “data representation”. From past examination paper responses this appears to have been well understood previously. If the candidate has covered the specification content fully then it is hoped that students would be fascinated to learn how the same binary number can be interpreted in so many different ways: a basic machine code instruction to add two numbers together, a musical note, the colour of a pixel, one of the characters in a text string, one of several different number types etc.

## Q17.

It was pleasing to see that most candidates did very well on this question. Part (a) was often correct. It was also pleasing to see a greater proportion of candidates showing some understanding of the stored program concept.

## Q18.

(a) It was surprising to see how many candidates gave devices that were not on the list. This showed that they had not read the question clearly enough. There were few candidates who obtained full marks. By looking carefully at the diagram, it should have been clear from the direction of the arrows which buses were which. It should also be noted that ROM and RAM are abbreviations and are not guaranteed to obtain credit.

(b) Few candidates realised that it is the size of the address bus that limits the number of memory locations.

(c) It was pleasing to see that most candidates gave the correct response to this part.

## Q19.

(a) Good marks were obtained on this part of the question but there were a surprising number of candidates who could not state the full names correctly.

(b) The term hard disk controller was not well known. Candidates were more aware of the network interface card. This was one situation where a number of candidates failed to score by giving a brand name as their response.

(c) Another high scoring part although a large number of candidates only obtained two marks. Both the data and address buses were well known but the control bus was often mistakenly given as either a system bus or a serial bus.

(d) This was not answered well. Candidates often failed to obtain credit by weak description, for example “memory” could apply to a number of different methods of storage. It was important that the candidates made it clear that programs must be in the main memory from where they will be fetched and executed by the processor.



(e) Parity concepts do not seem to be well known by the candidates. Few candidates were able to explain what is meant by even parity and even fewer were able to explain how the computer system might use the parity bits.

## Q20.

(a) The names of the three buses were generally known but fewer candidates were able to explain how they operate. A bus is a device that carries signals around the system. Common misconceptions were that buses store data or send data. Incorrect answers included reference to system buses and/or memory buses and many candidates failed to obtain full credit by not expressing themselves well enough.

(b) The network adapter was well known but candidates failed to obtain full credit by being unable to express themselves adequately when explaining its purpose. The question stated that the network adapter connects the computer system to the local area network. An answer that restates this is not going to obtain any credit.

(c) Many candidates understood that faster transmission can be obtained by using parallel transmission. Some simply stated a definition of parallel communication

being many bits simultaneously transferred but failed to state why this might be of benefit.

(d) A substantial number of candidates stated that serial transmission was faster. Many more stated that it was cheaper but failed to give any reason as to why this should be the case. Many marks were lost on this part of the question.

## Q21.

Many candidates scored well on this question. However, a significant number did not take account of the arrow heads into and out of the system, and VDU controller and keyboard controller labels were interchanged.

The stored program concept generally is very poorly understood. The machine code instructions (program) and the data are stored in main memory and instructions are fetched sequentially and executed by the processor. Programs in main memory can be replaced by other programs at any time. Many candidates wrongly stated that this concept means that programs were stored in Read Only Memory or in secondary storage. It seems that many candidates are not clear about the difference between secondary storage and main memory, even when they had correctly labelled the diagram earlier in the question.

## Q22.

(a) This being almost a multiple-choice question meant that nearly all candidates picked up some marks. The majority of candidates did not pick up on the significance of the uni-direction arrow from the processor to the address bus as the major clue.

(b) This part was very poorly answered with the usual 'processor processes data' being a typical response in (i). Students who have knowledge of von Neumann architecture know that instructions have to be in main memory at the time of the fetch-execute cycle, though others could have picked up the mark by stating that the main memory is a temporary store for programs/data. Many candidates lost even that possibility by stating that it stored 'information'. In Computing 'information' is not a synonym for 'data'.



Part (ii) saw hardly any credible answers. Interrupts, clock signals and memory read, memory write were the usual correct answers.

Part (iii) saw a few correct answers with some losing the mark by stating the too vague 'location' rather than the more specific address, or commands rather than instructions.

## Q23.

(a) The clues were all there in the diagram to help candidates correctly identify the six numbered components. Many candidates managed to match up all six correctly. The secret was in the direction of the arrows to/from the component to the bus. Sadly, some candidates would not use the component names given in the question stem and invented their own. Using 'keyboard' instead of 'keyboard controller' was not given credit.

(b) A huge variety of answers, with 256 and 99999999 vying to be the most popular incorrect answers. Many candidates did not appreciate that the largest binary number which can be transmitted along 8 lines is 11111111, which is 255 in denary. Some candidates do not seem to understand the term 'denary'.

(c) It seemed that many candidates couldn't believe that the same answer was required for both parts thinking that there must be a catch. Some solved the dilemma by saying RAM and 'main memory' which gave them both marks; others said RAM and ROM which was unlucky. Yet others decided on virtual memory which is a memory management technique rather than actual memory. Candidates need to appreciate that the processor cannot access instructions or data directly from hard disk but that both need to be loaded into main memory prior to execution.

## Q24.

This focused on device controllers and buses. Candidates did not seem to appreciate that the voltages required for the correct operation of devices were different from those required by the processor. Thus, without a device controller, the processor would have to be re-designed every time a new device was installed. The use of device controllers can also cut down on the number of ports required as one controller can control more than one device of the same type.

The reason why the address bus only carries addresses in one direction is that the processor has to send the required address to the controller or to main memory for both the read and write operations. No data or signals have to return via the address bus from the device to the processor. The knowledge of typical signal lines in the control bus was patchy.

## Q25.

Part (a) was well answered, with only a few candidates not taking note of the direction of the arrows in the diagram. Part (b) rarely gained more than one mark. A significant number of candidates thought the stored program concept only referred to programs held in ROM. Many candidates thought programs are run from disk. Those who did say they were stored in main memory gained a mark, but often did not gain the mark for fetching and executing. Many candidates missed a mark by saying "stored in memory" rather than "main memory". In parts (c) and (d) candidates used the word "data" rather than "single bits". A common misconception was "serial data can go one way and parallel can go both ways". Part (d) was either answered really well or poorly. Candidates need to appreciate that parallel transmission deteriorates over distance and therefore can only be used between devices in close proximity. In part (e) there did not seem to be a great understanding of asynchronous transmission and the need for synchronisation. Many candidates did not appreciate that just one character is being transmitted between the start and stop bits, just referring to data. In asynchronous data transmission when no data are being sent the signal transmitted represents 0. This ensures that the first signal received is always a change from 0 to 1. This change in voltage can be used to start the clock of the receiving device. The receiver will then read the 8 data bits. The stop bit ensures that the receiving device has time to recover and the next start bit will be recognised.

## Q26.

A high scoring question but few candidates obtained all three marks for part (c). Many answers stated that more memory speeds up the machine but no understanding as to how or why this might be the case.