7.2 The stored program concept Mark Scheme

Q1.

(a) A set of rules / regulations (to allow communication between devices) // set of agreed signals / codes for data exchange;
NE a rule // a regulation // a signal // a code
NE instruction(s)

1

(b) Analyses statement by statement each line of source code
A runs / translates / executes line by line
R compiles (line by line)

Calls routines to carry out each instruction / statement

Max 2

(c) Instructions / programs stored (with data) in main memory; **A** memory // RAM

Program run by fetching, (decoding and executing) instructions
(from main memory)* in sequence;

Program can be replaced by loading another program into (main) memory

Contents of a (main) memory location can be interpreted as either an instruction or data;

* = This mark can be awarded without the explicit reference to main memory if main memory has already been mentioned elsewhere in the response.

Otherwise, the answer must make clear that the instructions are coming from the main memory to get this mark.

3

(d) LOAD 21
STORE 23

LOAD 22
STORE 21

LOAD 23
STORE 22

1 mark for value from 21 stored into 23;
1 mark for value from 22 being moved to 21;
1 mark for value from 23 being moved to 22;

Alternative :

LOAD 22
STORE 23

LOAD 21
STORE 22

LOAD 23
STORE 21

1 mark for value from 22 stored into 23;
1 mark for value from 21 being moved to 22;
1 mark for value from 23 being moved to 21;
DPT if a different temporary storage area is used
I end of statement separators
Max 2 if the program does not fully work

3

(e) Robots find it hard to adapt to changes in environment // Robots are unable to adapt to changes easily;

Robots find it hard to work with 3D vision;

Robots find it hard to detect edges between similar objects // robots find it hard to perform shape detection;

Robots find it hard to get feedback when gripping items;

Robots find it hard to pick up balls // ball difficult shape to grip // balls can roll away;

Robots have limited processing power // too many variables to deal with;

Programming for vision/grip is a complex problem;
A child builds up experience of using touch / vision;

A Robot cannot recognise when it makes mistakes;
A Robot can't think for themselves // can't perform lateral thinking

Max 3

(f) (i) (Lens focuses) light / photons onto image sensor;
R if uses 'reflection'

Image sensor is a CMOS / CCD / photoelectric device;
CCD used ADC to convert measurement of light intensity into binary;
CMOS uses transistors to generate binary value;
Image sensor converts light into discrete / electrical signals / binary;

Image is captured when the shutter is pressed;
Large pixels collect more electrons than small pixels and so produce better quality images;
Firmware performs data processing to "tidy up" image;
(Colour) filter used to generate data separately for Red, Green, Blue colour components;
Aperture / shutter speed can be adjusted to cope with varying lighting conditions;
Image is recorded as group / array of pixels // Image sensor consists of array of pixel (sensors)//etched into the image sensor's silicon are pixels;

Image data transferred to robot;
Image data usually stored on solid-state disk;

Max 3

(ii) Robot has a low powered microprocessor;

Too much image data for the robot to process quickly // smaller resolution can be processed quicker;

A high resolution image has too much image data for the robot to store // low resolution uses less storage space;

Do not need high resolution to determine colour of balls;

NE allows more images to be stored

Max 1

[16]

Q2.

(a) (memory) address / location;
R Line number

1

(b) second (generation) //assembly language/code/program //
2 / 2nd;

1

(c) (i) assembler;
R. assembly

1

(ii) error list / error report / error count / **A** error message / highlight statement(s) illegally formed / instruction count // symbol table;
R error

1

(d) program (instructions are) transferred from backing store to main memory;
program consists of a sequence of instructions;
stored in a (continuous area of) main memory;
an instruction is fetched (and decoded);
and then instruction executed (by the processor);
program can be replaced by another program at any time;
program instructions are treated as data;

Max 4

[8]

Q3.

(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 ;

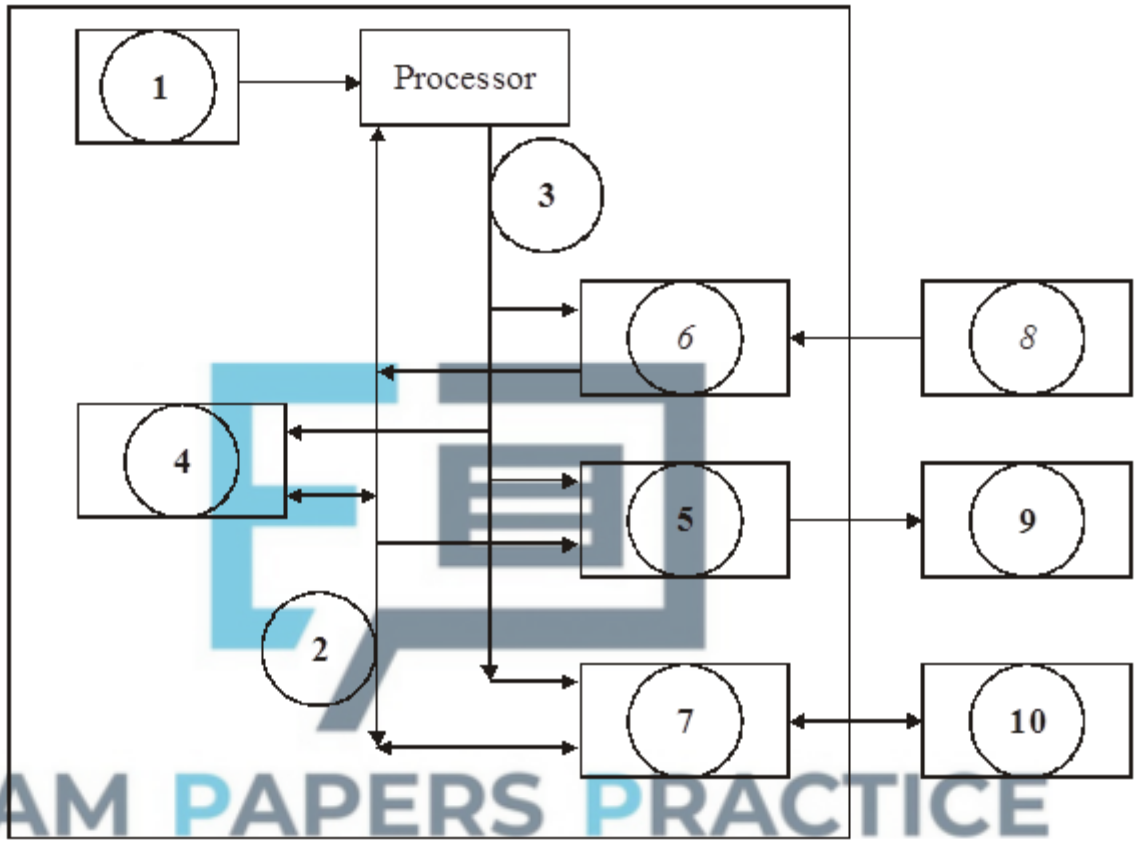
Program consists of a sequence of instructions ;
Program is stored in main memory ;
and can be replaced by another program at any time ;
Instructions are fetched (in sequence) ;
Decoded ;
and then executed ;

Max 3

[9]

Q4.

(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]

Q5.

(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;

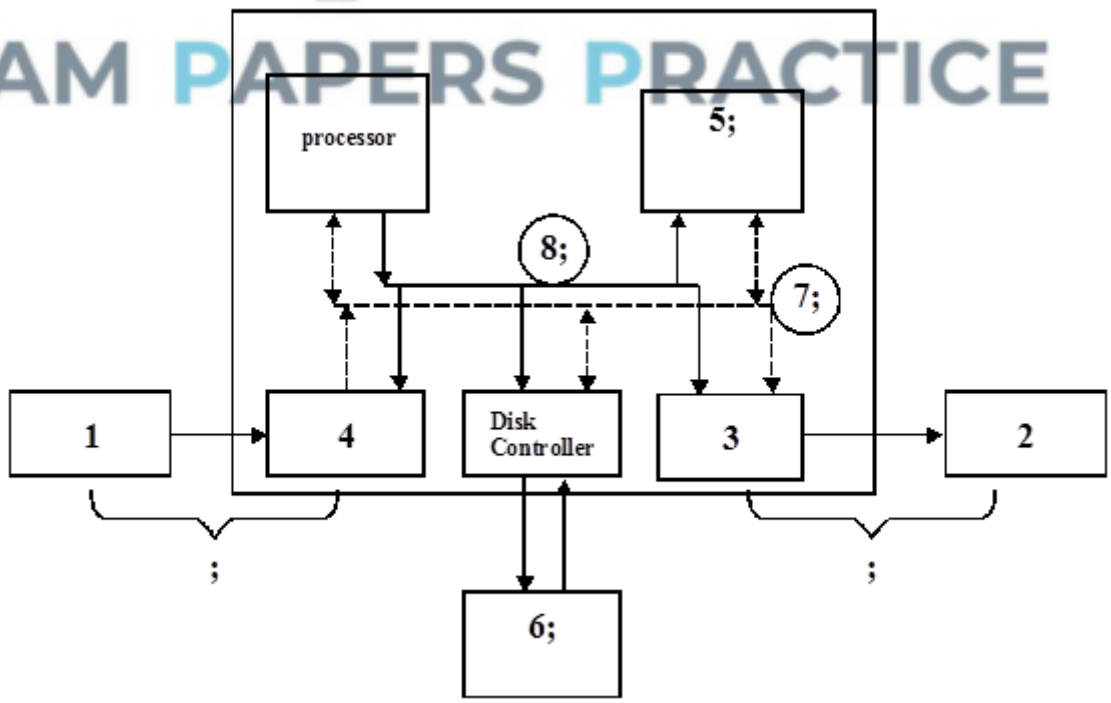
Any 2

2

[12]

Q6.

(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]

Q7.

(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

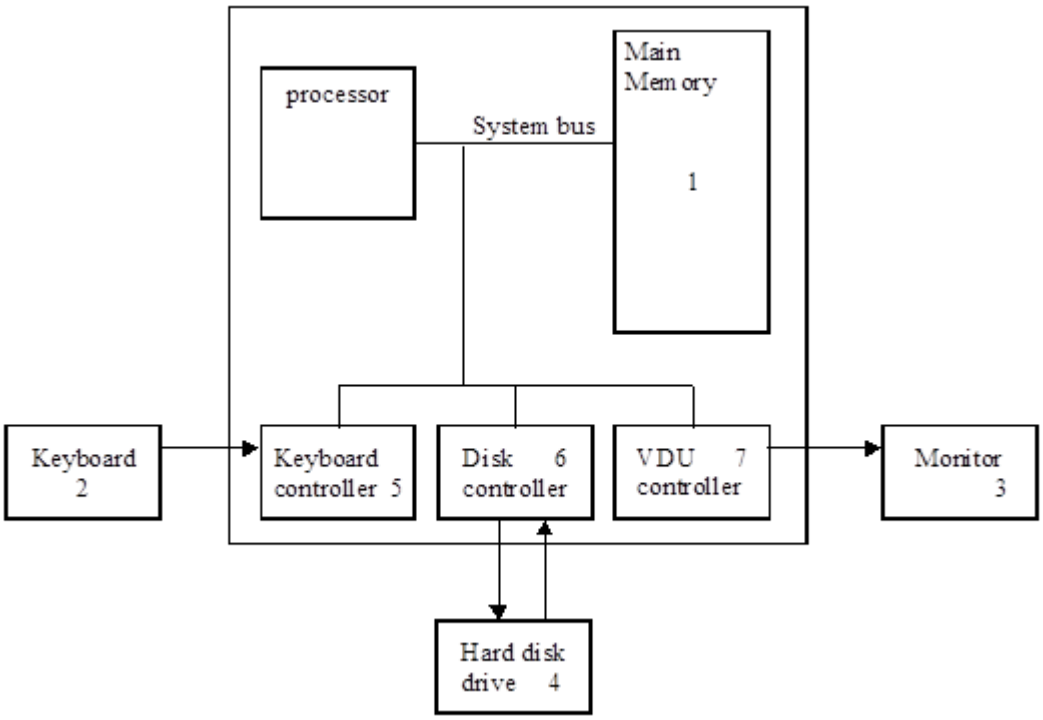
(ii) Main memory / primary memory / RAM /
Immediate Access Store / IAS;
A cache memory;
A registers;
R processor/ CPU
R hard drive
R memory

1

[9]

Q8.

(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]

Q1.

This question asked students about a variety of topics all linked back to the idea of robotics. Over half of all students correctly provided a definition for protocol and the clearer answers linked this to the idea of an agreed set of rules to allow communication between devices. Some students who failed to secure the mark answered along the lines of instructions and programs rather than the idea of communication.

Part (b) asked students to identify how a HLL interpreter works. It was perhaps surprising that only half of the students managed to secure at least one mark on this question. It is clear that students got confused with the differences between a compiler and an interpreter with, some students answering this question by stating that it would 'compile'. Answers that just stated that 'it would interpret code….' also failed to secure marks. How an interpreter works beyond just translating code line by line is clearly not well understood and perhaps is an area centres could be encouraged to look at further.

Part (c) asked about the stored program concept. As a topic included in the name of the examination unit it was surprising to see that less than half of the students secured a mark on this question part. Of the credit worthy points made, it was common to see the idea that instructions are stored in the main memory of a device. A few students then went on to correctly identify that instructions are then fetched and executed by the processor. It was pleasing to see some students then discuss that the stored program concept allows different programs to be switched in and out of memory providing the ability to run different programs.

Unit 2 looks at only three machine code instructions and these were all given on the question paper in part (d) as a reminder to students. The correct answer only needed use of the LOAD and STORE instructions and over half of all students secured all three marks for this question part. A common mistake was to just see an answer of the form 'LOAD 21 ADD 22 STORE 23' showing that perhaps a student did not understand what the question was really asking them to perform which was swapping two stored values around.

Part (e) was a question looking at the differences between robotics and how we cope with situations. It was pleasing to see students identify aspects such as robots finding it hard to get feedback when gripping an item, and the problems in separating two similar coloured balls when they are obscuring each other. An answer that was rarely seen was the idea of a child building up experience over time and learning, compared to a robot just being programmed.

Students continue to struggle with identifying the major principles of how hardware devices work and it was common that no marks were achieved when discussing the digital camera and nearly 10% left this question part blank. Better answers considered items such as the shutter opening and closing to capture the image and correctly identifying the use of a sensor with more able students stating that this would be a CMOS or CCD device. It was surprising to see a few candidates talk about the use of film to capture the image.

The second part to the digital camera question was answered well and it was clear that students could link the idea of low resolution images to the needs of a robot. The simplest answer was to talk about the lower storage space required but the more able students linked this to the robot being able to process this amount of data faster or even that to identify colour differences would not require high resolution. Students should be encouraged to express their answers with direct reference to a question context, when appropriate, as this does allow them to demonstrate their understanding at a higher level.

Q2.

Parts (a) and (b) were generally well answered but less so (c) and (d). Many candidates gave a vague answer for the assembler output as 'errors' and this was considered insufficient.

Very few candidates were able to store the maximum 4 marks in part (d). There were very few convincing answers, and common misconceptions were that the program is executed from the hard disk, with no mention of program instructions. Often candidates generally described the need for the program to be loaded into main memory, without any further explanation.

Q3.

(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.

Q4.

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.

Q5.

(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.

Q6.

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.

Q7.

(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.

Q8.

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.