THE BCS PROFESSIONAL EXAMINATION
Certificate

April 2004

EXAMINERS' REPORT

Technology

General

Important: Candidates should be reminded to indicate questions attempted on the front cover of the answer book. A number of candidates also seem uncertain how this is to be done, i.e. circling the appropriate question numbers.

Candidates' performance in section A was reasonable for this sitting. The standard of the answers was fairly good across the centres. Candidates managed to score a pass mark in most of the questions. In contrast to October 2003, candidates who chose questions carefully scored well

The most popular question was question 1 and the least popular was question 3.

Some candidates did not encircle the questions' number at the front of their scripts. The level of English was bad in some centres, hence the inability for candidates to express themselves clearly.

Overall the marks attained in section B are significantly lower than the previous years. In many cases candidates did not pay attention to the requirements of the question, such as "give examples", leading to low marks on such questions. It appears that candidates are anticipating questions, from previous papers. While past papers are a good indicator, technology changes quite rapidly and candidates need to keep up-to-date with emerging issues.

Question 1

1. *a)* The central processing unit (CPU) of a computer is made up of *registers*, *buses*, one or more *ALU*s, and a *control unit*. Explain the role of each of the elements in italic font in a CPU. **(10 marks)**

b) Draw the diagram of a basic von Neumann computer at the bus/register/ALU level. **(5 marks)**

c) Using the diagram in part *b)*, explain how an instruction is read from memory and executed. **(15 marks)**

Answer Pointers

a)

Registers are special-purpose memory elements that hold a single item of data, usually a word (the wordlength is the length of the basic element of data processed by the computer). Registers are part of the CPU and can be rapidly accessed (they are not in memory and don't require an address to access them). User-visible registers (the PC, status register, data registers, address/index/pointer registers) can be accessed by the programmer/compiler via machine-level instructions. Invisible register such as the MAR, MBR, IR are used by the processor for internal house-keeping and are not directly accessible by the programmer.

Generally, a register holds temporary working data. Registers are also required by instructions such as ADD P,D0 (add the contents of memory location P to data register D3). By using internal registers, the instruction does not have to have a full 32-bit operand address.

A **bus** is a common data path (usually the same width as a word) that moves data between functional units and registers. Buses exist within the CPU and externally (between the CPU and memory and peripherals). Without a bus, it would be impossible to design efficient CPUs because data would have to go from point-to-point and that would require too many data paths.

The **ALU** (arithmetic and logical unit) performs calculations for the CPU. The three types of calculation are arithmetic (add, subtract, multiply, divide), logical (AND, OR, NOT, EOR), and shift (LSR, LSR, ASR, ROL, ROR). An ALU has two word-length inputs for the source operands, a control function input (select operation) and a word-length output. ALUs also have a carry-in and a carry out bit. Finally, the ALU has a flag register output that records whether the last operation was zero, negative, generated a carry, or overflowed.

The **control unit** is the most complex part of a processor. It takes the op-code bits from the instruction register plus a clock signal and generates all the signals necessary to cause the current instruction to be executed.

b)
See diagram on the next page. A student's answer may not be as detailed as this. The essentials are the PC, and address/data paths required to read an instruction from memory and pass it to the IR, and then send the operand address in the instruction to memory to read the operand. Data from the memory can be operated on by the ALU using data from a register. The result of the ALU can be fed to memory or to a data register.

c)
Suppose the next instruction to be executed is ADD P,D0 (add the contents of memory location P to the data register D0 and put the result in D0).

Initially, the address of the current instruction is in the program counter, PC. This address is copied to the memory address register, MAR, and the contents of the program counter sent to the incrementer to be incremented to point at the next instruction. That is;

[MAR] ← [PC]	Copy PC to memory address register
[PC] ← [PC] + 4	Increment PC (most instructions occupy 4 bytes)

The instruction is read from memory using the address in the MAR and transferred to the memory data register and then to the instruction register, IR.

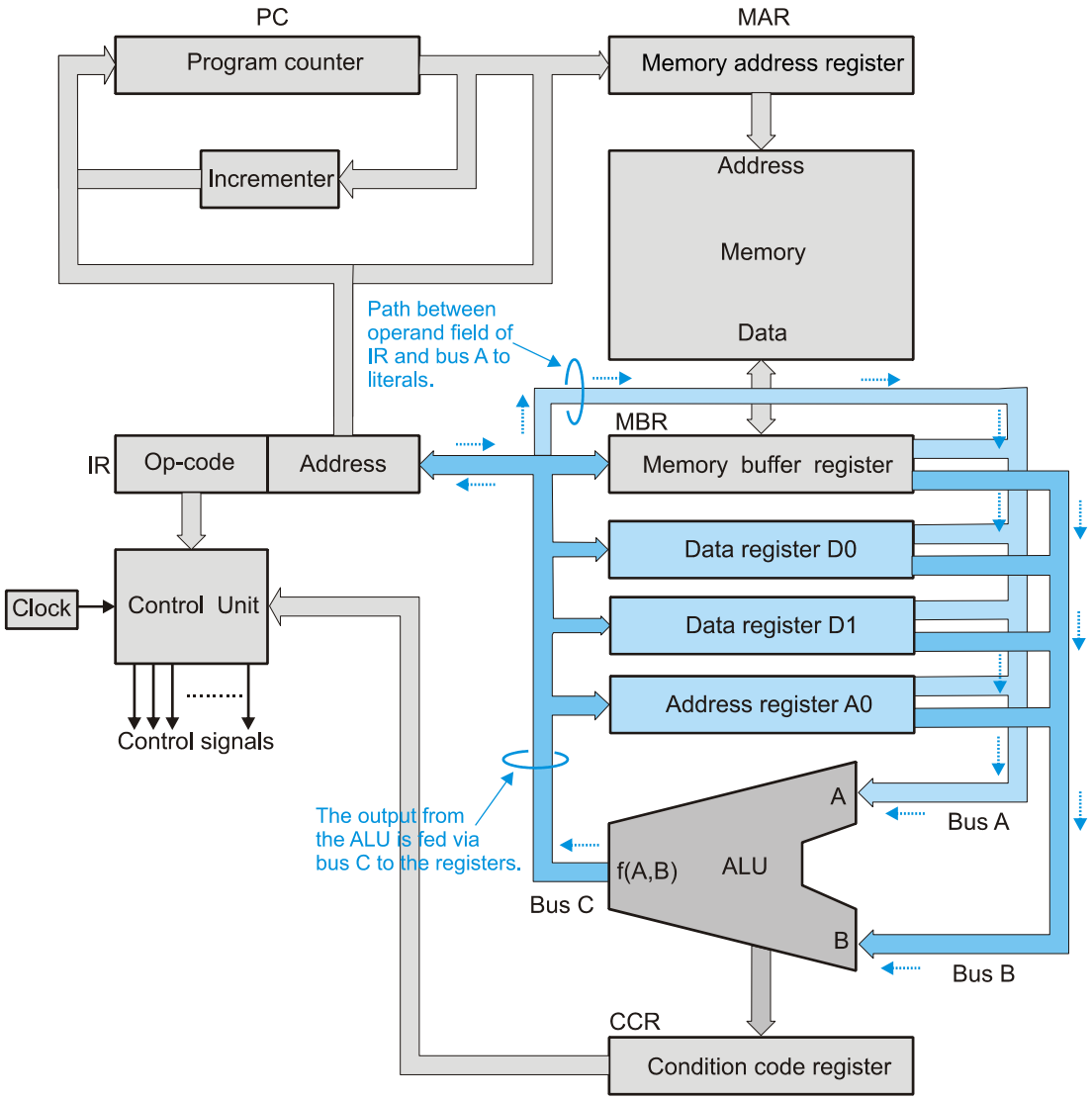
[MBR] ← [M([MAR])]	Read instruction at address in MAR. Copy to MBR
[IR] ← [MBR]	Copy instruction in MBR to instruction register

This sequence is common to all instructions. The instruction in the IR is then interpreted (i.e., executed). In this case, it is an ADD.

[MAR] ← [$IR_{operand}$]	Copy operand address to memory address register
[MBR] ← [M([MAR])]	Read operand at address in MAR. Copy to MBR

At this stage the memory buffer register contains the actual value of the operand..

ALU ← [MBR]	Send operand to ALU
ALU ← [D0]	Send second operand in D0 to ALU
Set ALU to addition	
[D0] ← ALU	Put result in destination register



Examiners' Comments

a. This section of the question was well attempted. On average, candidates scored 7 marks. Good definitions of registers and buses including examples were given. The role of these elements were also well explained. Candidates did not show a good understanding of ALUs and control unit. Answers, in general for these were, 'to control the various part of the computer' or 'where all arithmetic is performed'.

b. The diagram was well drawn. Most candidates included the PC, MAR and MBR. They also showed interaction of these with the memory. Candidates scored well in this section.

c. Section c of the question proved to be the most difficult. Very few candidates were able to explain the Fetch Execute Cycle properly. Those who used an RTL format were able to provide a sensible answer. Good answers included a step by step explanation of how the registers are involved in the cycle, with the PC being incremented, etc. Candidates who adopted a more essay type answer were simply unable to explain the various steps of the cycle. It was really unfortunate that candidates were not well prepared, as this section of the question carried most of the marks.

Question 2

2. A computer transfers a block of data between its main store (i.e., memory) and its hard disk drive.

a) Describe the role of a typical operating system in this data transaction. **(10 marks)**

b) The data transfer may start when the disk requests a transfer by generating an interrupt. Explain the role of interrupts in I/O transactions and describe the sequence of events that might take place when the disk interrupts the computer. **(10 marks)**

c) Data may be transferred from the main store to the disk by means of DMA. Explain the meaning of DMA and how it is used to transfer data to (or from) a disk drive. **(10 marks)**

Answer Pointers

a)

Data is physically transferred by special purpose devices (disk controllers and DMA controllers). These are controlled by systems level software that is normally part of the operating system.

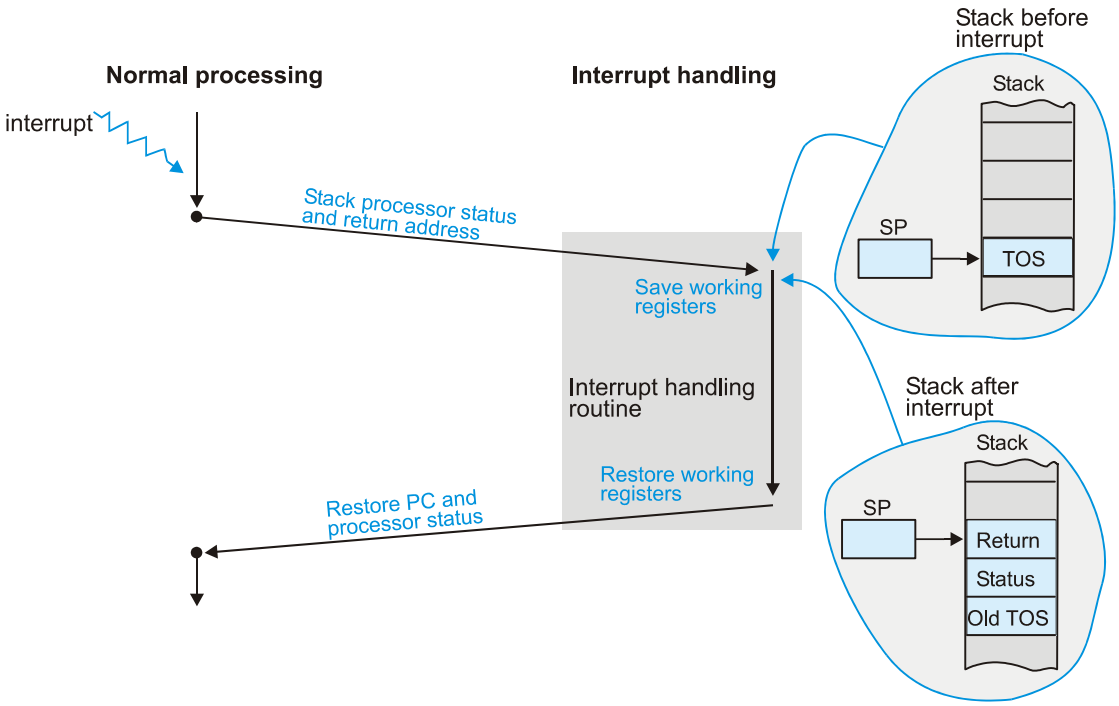
The operating system forms an interface between the user's application and the underlying system software. This approach is used to stop the user accessing I/O hardware directly and possibly crashing the system or causing unintended side-effects.

The user requests an I/O transaction from the operating system (normally by a TRAP or software interrupt). The operating system takes over control and sets up the data transfer using parameters supplied by the application. For example, the OS software may set up the DMA controller and disk controllers and request the data transfer from the hardware. The OS my then return control to the user application while the transfer takes place in the background. When the operation has been executed, the disc controller/DMA system may interrupt the processor and the OS is called again. The OS then communicates with the application informing it that the transfer is complete.

The operating system controls all I/O devices and ensures that the user cannot access hardware directly. The OS also deals with faulty hardware, and corrupted data.

b)

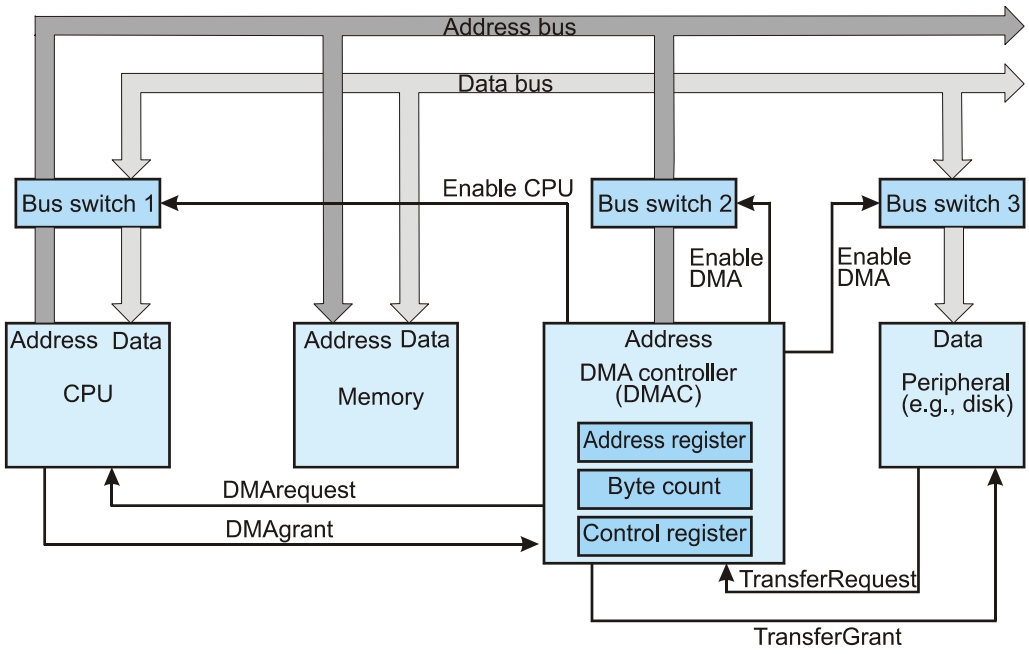
An interrupt is a request for service from a device (or a process). Interrupts allow the computer to intervene in an operation only when an input or output device is ready. Typically (see diagram on the next page) normal processing is taking place and a device such as a DMA controller interrupts (sends a hardware request to the CPU for service on an interrupt request line). The CPU saves its context on the stack (processor status word and program counter) and jumps to an interrupt handler in the operating system. The context is the information required to resume pre-interrupt processing.



The interrupt handler performs the requested operation (in this case it might be to set up the DMA controller for a data transfer). Once this action has been completed, a return from interrupt is made; the saved machine status is returned and the CPU continues exactly as it was at the point at which the interrupt occurred.

c)
DMA is a mechanism whereby data is transferred directly between a peripheral such as a disk drive and memory without passing though the processor. Essentially, a DMA controller takes over the computer's bus and controls all data transfers.

The diagram on the next page illustrates the DMA mechanism. The bus switches connect the CPU, DMA controller, and peripheral to the data bus. In normal operation the CPU accesses the buses. The DMA controller requests the CPU to give up the bus and turns off the CPU's bus switch and turns on the DMA controller bus switch. The DMA controller provides address and read/write signals for the data transfer (normally data is transferred into susceptive memory locations) .



Examiners' Comments

a. Answers were very general. Although candidates seemed to know what the operating system is, very few described what the operating system does. Some candidates mentioned issues such as memory management, job scheduling, etc. However, these were neither described nor applied to the data transaction.

b. Most candidates were able to define what an interrupt is. Good answers included a thorough description of interrupts and how they applied to the given scenario. Weaker answers simply mentioned the various types of interrupts.

c. This was the least popular section of this question. Most candidates managed to define DMA stating that the CPU is not involved. Very few candidates were then able to use this concept in explaining how DMA controller takes control of the data transfer. Some candidates produced interesting sketches/diagrams which showed an understanding of the technique.

Question 3

3. "Errors" can arise in a computer from a variety of causes. Such errors may lead to loss of data, incorrect operation, or even system crashes.

Describe the types of error that can occur in a computer and explain how they can be dealt with and how the error can either be corrected or the system prevented from crashing.

Note: In this context we exclude errors of accuracy and precision in the sense they are used by numerical mathematicians. **(30 marks)**

Answer Pointers

NOTE – this is a general question and credit should be given to students who answer this wide-ranging question sensibly.

Errors in a computer arise from several sources. Some of these are:

Data errors: A data error can arise when information is corrupted. This happens when data is received from inherently unreliable sources such as communication links and disk drives or CDs.

These data errors are normally dealt with by encoding the data using redundant information (an error correcting code). Recovery can be made by using error detecting and correcting software or hardware.

Data errors can occur during certain arithmetic operations. For example, an attempt to divide by zero yields infinity which a computer cannot deal with. Some processors generate an exception when division by zero is attempted (an exception is an operating system call). Data errors can take place when numbers overflow (i.e., a result is larger than the number of bits used to represent it). Errors can also occur when the subscript of an array is incorrectly calculated and an element is accessed outside the arrays bounds.

Software errors: There are several types of error that occur during the execution of code.

i. Illegal instruction op-code. If an illegal op-code (one that is not part of the instruction set) is loaded into the instruction register, an exception is raised and the operating system called. An illegal instruction may be cause by corrupted code, but it is usually the result of an incorrectly executed computed goto when the target address is a block of data rather than code.
ii. Privilege violation. Some computers operate at different levels of privilege; for example, the operating system has the highest level of privilege and may execute all instructions. A user level (applications program) has a lower level of privilege. This mechanism prevents the user performing inappropriate operations (operations that could crash the system or corrupt another user's code or modify the operating system).
iii. Memory violation. In systems with memory management, memory is partitioned between useds and the operating system. Memory management ensures that one user cannot access (without permission) the memory space of another user.

Examiners' Comments

This question was the least popular one among candidates. Those who attempted this question did not produce good answers. Very few answers focused on errors 'that can occur in a computer'. It was also difficult for most candidates to explain how errors can be prevented or corrected. Some candidates provided answers related to errors encountered during program development. They described in details syntax and semantic errors.

Question 4

4. Compare and contrast the inkjet (bubble jet) and laser printers.

Your answer should cover their principles of operation, durability, reliability, and operating costs. You should also explain the type of applications for which each printer is best fitted in both the home and small office.

(30 marks)

Answer Pointers

This is a fairly open-ended question. I expect diagrams of the basic operation of both an inkjet printer and laser printer.

The key features are:

The **inkjet printer** used a print head that squirts a tiny drop of ink onto the paper. The drop is ejected from the print head either by heating the ink and momentarily boiling it, or by flexing a quartz crystal to send a shockwave through the ink to eject a drop. A print head has multiple nozzles.

The **laser printer** uses a drum coated with a photoconductive material. First a high voltage charge is placed on the drum. An image is projected on the drum by a laser (or other means) which causes the charge to leak away in light areas. The drum is then covered with a toner (a fine

powder) which sticks to the charged (dark) areas. The drum is then rotated against the paper and the toner transferred to the paper where it is set (fused) by heating.

Both ink jet and laser printers are fairly reliable. However, the inkjet printer is normally used for domestic and small office applications with small print runs. Inkjet printers operate a line at a time and scan the paper by moving the print head along a line. This means that inkjet printers are ultimately slower than laser printers that print an image at a time.

Inkjet printers are remarkably cheap. This is partially because they are mass produced and partially because the complexity of the printer is in the print head which is thrown away when the ink is used. This approach means that the life of the most critical part of the printer need be measured only in terms of hundreds of sheets of paper rather than tens of thousands. Moreover, one problem of inkjet printing is closing – the nozzles become clogged by dried ink and fail. By replacing the print head when the ink is used, no maintenance is required.

The laser printer is available in low-cost domestic versions that replace the drum and the toner when they are exhausted after about 5000 sheets. This is a similar approach to maintenance that the inkjet takes. High volume professional laser printers still use a separate toner and many of the moving/operating parts are replaceable.

Inkjet printers can achieve photorealistic prints (that is, of a quality as high as conventional colour photography).Resolutions up to 4,000 dots per inch are possible. The resolution of laser printers is lower than that of inkjets (a high volume laser printer might achieve a resolution of only 600 dots per inch and a high resolution laser printer might achieve only 1200 dots per inch). Colour laser printers are relatively expensive because the print mechanism has to be (partially) duplicated for each of the primary colours plus black.

Examiners' Comments

This was the second most popular question. In general, candidates attempted this question fairly well. Candidates scored well in this answer. The style and approach of the answer varied. Some candidates adopted an essay type answer and discussed in details both printers, ensuring that they covered the issues that were mentioned in the question. The other approach was to use a table made of two columns, one for each printer. A comparison was then made between the two printers.

Question 5

5. *a)* Briefly explain the difference between machine language and assembly language. **(6 marks)**

b) An instruction includes an "op-code" and an "operand". What is the significance of each term? **(6 marks)**

Answer Pointers

a) **Machine language** is the lowest-level programming language and are the only languages understood by a computer. Machine languages are written as binary numbers and therefore very difficult for humans to understand. An **assembly language** contains the same instructions as a corresponding machine language, but the instructions have names (mnemonics) instead of binary numbers. There is a one-to-one correspondence between assembly and machine languages. Programmes written in assembly language are converted to machine language using an "Assembler". Every CPU has its own unique machine language.

b) **Op-code** describes the type of instruction to be performed including the addressing mode. **Operand** defines the variables to be used with the instruction, which may be a number, a memory reference or register. E.g an instruction in a hypothetical assembly language: LDA 101 would

mean load the contents of memory location 101 into the ALU. LDA is the op-code; 101 is the operand.

Examiner's Comments
In a) most candidates seemed only to know that machine language consists of binary numbers and assembly languages of mnemonics. The relationship between the languages needed to be explained fully.
In b) most candidates knew what op-codes and operands represent. This part of the question was answered well.

Question 6

6. *a)* What is meant by an overflow in binary arithmetic and how does it occur? **(4 marks)**

b) Carry out the following sums using 7 bit binary two's complement, and state whether an overflow occurs or not, and why.

i) 65 + 72 **(4 marks)**
ii) 123 - 65 **(4 marks)**

Answer Pointers

a) An **overflow** occurs when the addition of two numbers results in a sum that cannot be contained in the computer's word size. An overflow is detected when two numbers of the same sign (positive or negative) result in a sum of the opposite sign. Overflow errors can occur without a "carry".

b) i) 65 = 01000001
72= 01001000

sum 10001001
An overflow occurs: the sum of two positive numbers is negative.

ii) 123 = 01111011
-65 = 10111111

sum 00111010
Carry occurs, but can be ignored. No overflow, the sum is correct sign.

Examiner's Comments
Most candidates did know that an overflow error occurs when the result cannot fit in the word size. However, this was invariably associated with their being a carry bit. Only a few candidates were aware of detection of the error by checking the wrong sign of the sum. Many candidates could not determine the two's compliment of a negative number (-65). Some candidates, on the other hand, added the numbers in decimal and then converted the sum to binary, for which no credit was given.

Question 7

7. In computer data handling, explain each of the following:

i) Character
ii) Byte
iii) Word
iv) File
v) Record
vi) Field **(12 marks)**

Answer Pointers

i) **Character** – any symbol that can be stored or manipulated by a computer, e.g. number, punctuation, letter. Usually represented by a byte.
ii) **Byte** – a group of 8 bits
iii) **Word** – the number of bytes (bits) that can be processed as one; the representation of numbers and instruction. 16-bit, 32-bit, etc.
iv) **File** – a collection of logically related data. Almost all information stored in a computer is in a file.
v) **Record** – a description of an entity, item or event; e.g. employee record may have "Name, Position, Salary, Department".
vi) **Field** – fields are basic components of a record.

Examiner's Comments

Most candidates knew most of the data types, and the question was answered well.

Question 8

8. Explain the following concepts in data communication, and give appropriate examples in each case:

i) Simplex
ii) Half-Duplex
iii) Full-Duplex **(12 marks)**

Answer Pointers

i) Simplex – communication in only one direction, where one is the transmitter and the other is the receiver. Examples: radio, TV broadcasts
ii) Half-duplex – communication in one direction at a time. Walkie-Talkie.
iii) Full-duplex – communication in two directions simultaneously. Telephone.

Examiner's Comments

Most candidates who attempted this question knew the correct answers.

Question 9

9. *a)* Explain how a stack can be used to support the subroutine call/return mechanism and the passing of parameters. Use examples to illustrate your answer. **(6 marks)**

b) Briefly describe the operation of virtual memory. **(6 marks)**

Answer Pointers

a) A **stack** is a last-in first-out list. Items are pushed onto the list or popped from it. In sub-routine calls, the current values in the registers, including the program counter, are pushed onto the stack. The address of the sub-routine call is then loaded in the program counter, so that program execution continues with the sub-routine. When the sub-routine call ends, the items in the stack are popped back into the registers, so that execution continues at the point before the sub-routine call.

In evaluation of expressions, the expressions are given in postfix form(reverse Polish), e.g. a+b is given as a b +. Expressions are scanned from left to right. If the element is a variable or constant, they are pushed onto the stack. If the element is an operator, pop the top two items from the stack; perform the operation and push the result back onto the stack. This can be used to evaluate expressions of any complexity.

b) **Virtual memory** is used in multi-programming or to run programs much larger than available physical memory. Secondary memory is divided into pages of equal size while the physical is also divided into frames of the same size. The concept of demand paging is used: pages are loaded into main memory only when required. When a process references a page which is not in main memory, a page-fault occurs; this causes the required page to be loaded into main memory. Only a few pages of a running process need to be in memory, so many processes can be run at the same time. Different techniques are used to determine which page should be removed when a new page is loaded.

Examiner's Comments
a) While most candidates knew that a stack was last-in first-out storage, many could not explain how it is used in sub-routine calls, or in evaluation of expressions. This part of the question was very poorly answered.
b) Again, many of the candidates knew that virtual memory involves dividing available memory space into frames and pages, but, there was confusion between what frames and pages are. Most candidates could not clearly explain the process of demand paging.

Question 10

10. Explain the following concepts from network communication: token, physical topology, logical topology. Give examples as appropriate. **(12 marks)**

Answer Pointers
Token: In network communication, a token is a special series of bits that travels around a ring (or bus) network. The token includes control signals to indicate whether it is carrying a message or free. As the token circulates, senders capture a free token, add their message and set the status of the token to busy. The token travels round the ring to the addressee computer which copies the message and returns to the sender. The sender finally places a free token on the network. There is only one token for each network, so there is no possibility that two computers will attempt to transmit messages at the same time.

Physical topology: the structure or arrangement of the devices, through the cables, in a network. Draw examples of bus, star, ring topologies.

Logical topology: the way that the data passes through the network from one device to the next without regard to the physical interconnection of the devices. The logical topology is determined by network protocols. The Ethernet is an example of a logical bus topology; its physical topology is usually star.

Examiner's Comments
Most candidates had an idea what tokens are used for, although they seemed to associate exclusively with the token ring network. Similarly, most candidates could differentiate between logical and physical topologies, but, could not definitively explain what logical topology refers to. The question was moderately well answered.

Question 11

11. Describe the following terms related to computer processing: deadlock, pipeline, multi-tasking, parallel processing. **(12 marks)**

Answer Pointers
Deadlock occurs when two processes are each waiting for the other to complete (or release a resource) before proceeding. The result is that both processes hang. Deadlocks occur most

commonly in multitasking and client/server environments. The operating system can resolve the deadlock, but this doesn't always happen.

Pipeline refers to processing of the different stages of an instruction cycle in consecutive clock time-interval. The instruction cycles has three stages: fetch, decode and execute. Thus, in one clock time-interval an instruction is fetched, then it is decoded in the following time-interval and executed in the third interval. This increases throughput in single processor systems.

Multi-tasking refers running different processes (programs) with a single processor by time-slicing between the processes. The operating system performs the scheduling of the processes so it appears like the programs are running concurrently.

Parallel Processing refers to the use of two or more processors in computer systems to run one process or different processes concurrently. Each of the processors may have its own memory, but the operating system has to carry out more sophisticated resource sharing.

Examiner's Comments
Many candidates could not explain what deadlock or pipelines are. There was also confusion between multitasking and parallel processing. This could be due to the fact that some sources use multitasking and multiprocessing interchangeably. The question was answered moderately well.

Question 12

12. The following are services available on the Internet. Describe each and comment on their use.

i) FTP
ii) Chat Rooms
iii) SPAM
iv) Telnet **(12 marks)**

Answer Pointers

i) FTP - short for file transfer protocol. Uses the TCP/IP protocol to transfer files (download/upload) files across the Internet. Can be used to upload files to remote Web server.

ii) Chat Rooms - offer real-time communication between two users. Once a chat session is initiated, text typed by one user appears on the other user's monitor. There are now negative effects of its use by people who pretend to be what they are not, with criminal intentions especially to children.

iii) SPAM – electronic junk mail, usually, advertising something. This is now a real nuisance and generates unnecessary traffic on the Internet.

iv) Telnet – a terminal emulation programs which enables one computer to connect to another remote computer over the Internet. Once logged onto the remote computer commands entered on the local computer are executable on the remote computer. Can be used to control a remote Web server, for example.

Examiner's Comments
This question was meant to test the candidates' knowledge of contemporary issues in computing technology. Most candidates were familiar with the issues, although it was expected that they would also comment on their usage, especially Chat Rooms and SPAM. The question was moderately well answered by most candidates.