THE BCS PROFESSIONAL EXAMINATION
Certificate

October 2004

EXAMINERS' REPORT

Technology

General

Candidates' performance in section A was average for this sitting. The standard of the answers was fairly good across centres. Candidates managed to score a pass mark in most of the questions.

The most popular question was question 2 and the least popular was question 1.

As in previous sittings, most candidates did not encircle the questions' number in the front of their scripts. Candidates failed to write the question number at the top of every page of the scripts. Centres and course providers must be made aware of this so that they can impress the rule on their candidates. The level of English was bad in some centres, hence the inability for candidates to express themselves clearly.

Many candidates did not expand on issues raised in their answers. If candidates had paid attention to the thirty marks available for every question, they would have produce enough materials to score the available marks. Simple statements or definitions did not allow candidates to score well.

It is also very unfortunate to note that many candidates and course providers did not analyse past trends for this paper. Some elements of the paper have been examined during previous sittings. If candidates had paid attention to this, they would have been better prepared and accordingly written good answers.

Question 1

1. *a)* A computer's architecture is defined by its assembly language instruction set. Assembly language, the human-readable form of a computer's machine code, performs primitive operations on data in memory locations or in registers.

Assembly language instructions specify an operation and one or more operands; for example, ADD 1234,D0 adds the contents of memory location to data register D0. The location or value of an operand is specified by means of an addressing mode. Explain the effect of EACH of the following three addressing modes and give examples of their application (use diagrams where necessary).

i) literal (or immediate) addressing
ii) direct (or absolute) addressing
iii) register indirect (or indexed) addressing **(10 marks)**

You must define the action carried out by each of the instructions you use in plain English. You are encouraged to use register transfer language, RTL, to define operations.

```
                                ;form inner product of X and Y
S = 0                           ;preset sum S to zero
For i = 0 to n – 1              ;REPEAT
S = S + y_i + x_i               ;  add pair of products to the sum
End for                         ;Until n products added to total
```

(15 marks)

c) Some computers provide an addressing mode called *relative addressing* or *program counter relative addressing*. Explain the operation of this addressing mode, its advantage to the programmer, and show how it might be used. **(5 marks)**

Answer Pointers

a)

In the LITERAL addressing the actual value of the operand is supplied by the instruction itself. For example, the assembly language level instruction ADD #1,D0 might be interpreted ass "add the value 1 to the contents of register D0"; that is,

[D0] ← [D0] + 1. Note that the "1" in the instruction refers to the value 1 and is NOT a reference to memory location with the address "1".

The purpose of this addressing mode is to reduce the number of memory accesses (the operand is part of the instruction and a separate memory fetch is not necessary to retrieve this operand). Literal addressing is used to implement constants whose values are known at assemble/compile time; for example, in

```
For I = 3 to 20
J = J *5
Next I
```

the constants are 3, 20 and 5. All these can be represented by immediate/literal addressing.

In DIRECT addressing (also called absolute addressing), the operand is expressed as a memory location; for example, ADD 1234,D0 means add the contents of memory location 1234 to the contents of data register D0; that is, [D0] ← [D0] + [1234].
This addressing mode is used to handle variables whose values change during the execution of a program.

In REGISTER INDIRECT addressing, the location of an operand is specified indirectly by means of a pointer in a register; for example, ADD (A0),D0 means add the contents of the memory location pointed at by A0 to register D0; that is,
[D0] ← [D0] + [[A0]].
Note two accesses are required to access the operand; register A0 is read to get the pointer and then the memory location pointed at is read to get the operand.
Because the contents of an address register can be modified, register indirect addressing permits the dynamic addressing required to access data structures such as tables, lists, vectors, matrices and arrays.

b)

Any code that looks like the following is acceptable. Note that I am looking for the ability to set up a pointer, access the data pointed at, and then increment the pointer.

```
    MOVE #0,D0          ;clear register D0 to act as the total
```

```
        MOVE #n,D1          ;set up loop counter in register D1 to max count
        MOVE #Y,A0          ;copy address of vector Y into pointer register A0
        MOVE #X,A1          ;copy address of vector X into pointer register A1

Loop    MOVE (A0),D2        ;get Yi (use address register indirect addressing) in D2
        MOVE (A1),D3        ;get Xi (Note A1 points to array X) in D3
        ADD  D2,D3          ;add Yi + Xi (reg D3 is the sum of D2 and D3)
        ADD  D3,D0          ;add Yi + Xi to running total in register D3
        ADD  #1,A0          ;point to next element in Y; that is, i = i + 1
        ADD  #1,A1          ;point to next element in X;
        SUB  #1,D1          ;decrement loop counter
        BNE  Loop           ;repeat until all elements operated on
```

Note that this is based on the 68K microprocessor. I have made a simplification (in practice we should write ADD #2,A0 – students need not know this subtle point). We are interested only is a student's ability to understand the use of register (pointer) based addressing.

c)
Program counter relative addressing is like register indirect addressing expect that the program counter is used as the base pointer. Few computers provide program counter relative addressing to access operands. The operating MOVE 400(PC),D0 is interpreted as "copy the data item 400 locations on from that pointed at by the program counter into register D0". That is, the location of an operand is specified by its distance from the current value in the program counter.

The advantage of program counter relative addressing is that it creates position independent code, PIC, because you do not have to recalculate operand addresses if the program is relocated.

Most computers provide program counter relative addressing for branch and jump instructions where the offset of a branch (eg BRA Look) is expressed as the number of bytes to branch with respect to the program counter. The advantage of relative addressing is that the offset (branch distance from current location) often need be only an 8-bit constant for typical loops and if-then-else constructs. Moreover, relative branching also facilitates position independent code.

Examiners' Guidance Notes
Section a) of the question was well attempted. On average, candidates scored 7 marks. Suitable explanations were provided for the various addressing modes. Some candidates used good diagrams to show how the addressing modes are applied.

Section b) was either very badly attempted or completely ignored. Very few candidates were able to produce assembly language codes. Hence, candidates who attempted this question were penalised as they lost fifteen valuable marks.

Candidates failed to notice that this section of the question was worth five marks only. Many candidates attempted to provide detailed description of relative addressing. However, answers were limited to the operation only.

Question 2

2. Few inventions have done more to promote the growth of computing than the Internet and the World Wide Web.

However, the very Internet that has done so much to help computing is beginning to suffer from problems caused by the spreading of *malware*. The word *malware* (**mal**icious soft**ware**) indicates software that has either a harmful effect (e.g., virus) or is a nuisance (SPAM).

Describe the various types of malware that are currently affecting computers and networks and discuss what steps the computer manufacturers, operating system designers and network implementers can take to prevent the spread of malware. **(30 marks)**

Answer Pointers

This is an open-ended questions and I am not looking for a specific answer. However, students should appreciate the range of malware currently in use (viruses, worms, Trojan horses, SPAM, popups, spyware and so on). I would expect students to mention one instance of each class (virus, Trojan horse, SPAM)

A good answer should discuss the principles of operations of some of the malware (explain how it works). Answers may even mention or discuss the exploitation of features such as "buffer overflow" in programs that allow data to be executed as code. Students should mention the problems of macros and scripting languages (i.e., the abuse of scripting features incorporated in modern applications software).

Students should discuss the approaches that can be taken to deal with malware; for example, by operating system designers preventing the execution of unauthorized code, by users taking greater care (not opening files from people you don't know), by third-party organizations selling anti-virus or anti-spam products, and by government legislation (the banning of practices that lead to the generation of malware such as SPAM).

A basic pass should be awarded to a student who knows of the problem and some of the typical solutions – but his or her level of knowledge is hardly above that of the non-specialist specialist lay-person. A high mark should be awarded to a student who is aware of the range of malware programs; knows how they are created and propagated, and knows (in some detail) how they can be prevented.

Extra marks can be awarded to students who have a sophisticated knowledge and include concepts such as stealth or polymorphic viruses (those that attempt to disguise themselves). Similarly, good students may also introduce modern malware such a "spyware" that infects a computer and reports the users surfing habits to a third party.

Examiners' Guidance Notes

Most candidates managed to identify the various malicious software such as virus, trojan horse and SPAM. Answers lacked description on how these malware affect the computers and networks.

Candidates provided general comments on the steps that need to be taken by the listed entities eg computer manufacturers. Answers lacked preciseness and clarity on what can help the users understand how to protect their systems against the spread of malware.

Question 3

3. Modern operating systems perform two functions. They provide a user interface, they control the system hardware and allocate resources. Explain how the operating system makes use of the following two hardware (machine-level) facilities. You should describe each of the two mechanisms and you should provide diagrams to illustrate your answer.

a) The role of interrupts (i.e., exceptions) **(15 marks)**
b) Memory management unit, MMU **(15 marks)**

Answer Pointers

a. An interrupt-driven system uses an interrupt request from a peripheral to request the processor's attention.

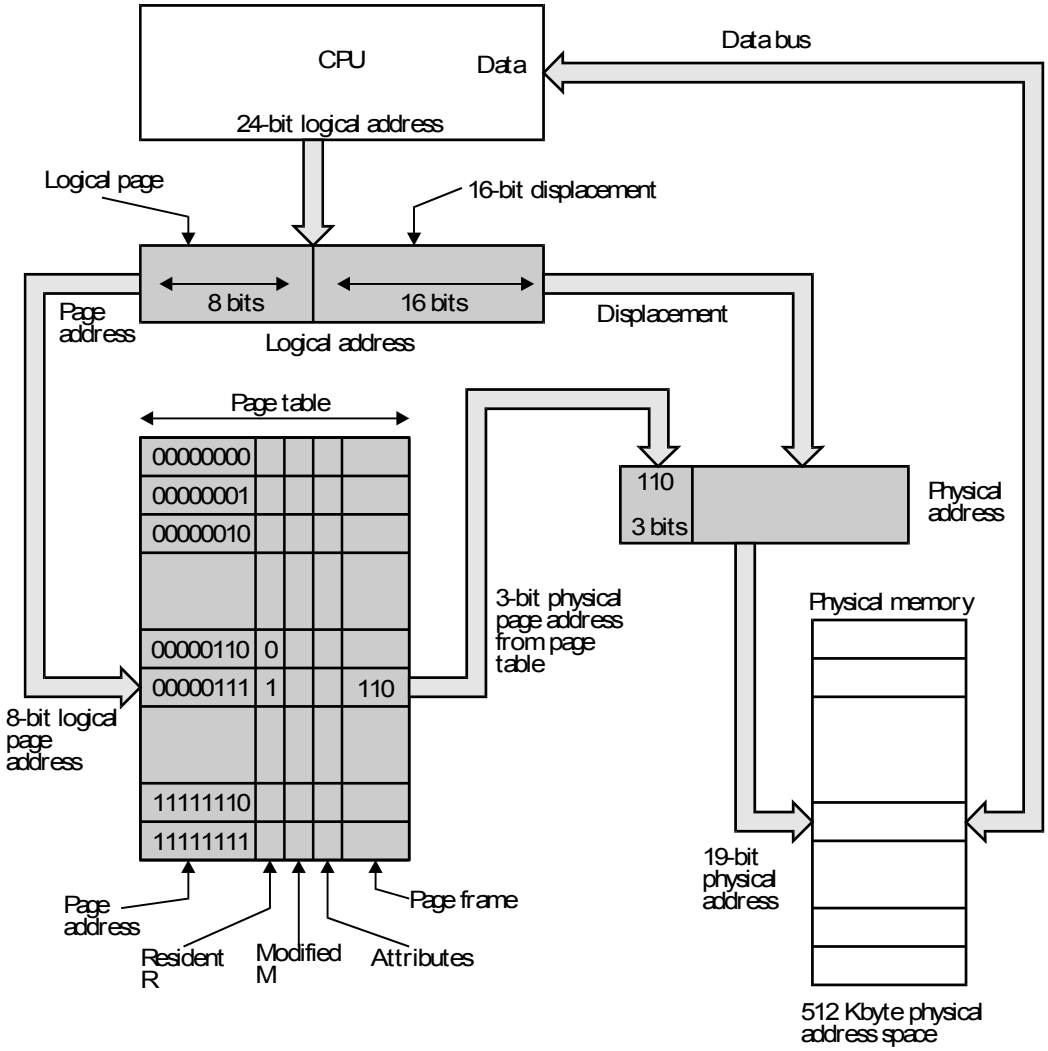
Typically, a peripheral signals an interrupt request via a special hardware signal path and:
1. The processor completes its current operation
2. If the interrupt mask is at a level the same as or higher than the current interrupt request, the request is ignored (masked)
3. If the interrupt is accepted, the current program counter and system status (carry, zero, negative flags etc) are saved (normally on the stack). The computer then determines the source of the interrupt (either by polling all the possible interrupting devices or by requesting a vector from the interrupter). Once the computer has determined the source of the interrupt, a jump is made to the appropriate interrupt handler. After the interrupt has been processed, the return address and status information is retrieved and the computer continues from the point at which it was interrupted.

Interrupts are used by the operating system for two basic purposes. One purpose is to recover from system errors such as failed memory accesses, bus time-outs, accessing memory that is not assigned to the current task. The other use of interrupts is in implementing I/O transactions. An interrupting device interrupts the processor when it is ready for a data transfer. The operating system uses the interrupt handler to perform the data transfer.

b. The operating system uses memory management to allocate physical address space to a computer's logical address space, to manage/integrate secondary storage devices, and to provide system security. Students may discuss concepts such a memory protection, virtual memory, memory sharing (e.g., one editor being shared by multiple users).

Without memory management programmers and compilers would have to provide the actual (i.e., absolute) addresses of data and instructions. With memory management, the operating system can map user programs anywhere within the system's RAM. A more important reason for memory management is the ability of the operating system to separate user tasks from each other and from the (protected) operating system. For example, if a page of memory is assigned to the operating system, any attempt by a user task to access the memory on that page will result in a page-fault. Some processors (CPUs) send out a special hardware signal when they are in their supervisor or operating system mode.

I expect students to provide a diagram that demonstrates an MMU of the form:



The key features of memory management are a look-up table to provide logical to physical address translation and a means of checking addresses for permissions (e.g., a user should not be able to access operating system address space or be able to write to read-only memory space).

Examiners' Guidance Notes
This question was not very popular among candidates, although it has been set previously.

a) Most candidates were able to define what interrupts are. It then proved to be difficult for candidates to take the discussion further and explain the role of interrupts. Some candidates provided examples of peripheral devices requiring the attention of the CPU. However, answers lacked clarity on the steps that both the device and CPU have to go through in order to deal with interrupts.

b) This section of the paper was badly attempted. Very few candidates were able to demonstrate that Memory Management Unit (MMU) exist. Some candidates attempted to describe Paging and associated issues. Others confused MMU with memory types and described in details RAM, ROM, Cache, etc.

If candidates were prepared for common computing techniques such as interrupts and MMU, they could have scored heavily.

Question 4

4. A high-performance computer such as a PC has a memory hierarchy; that is, it supports several memory subsystems, each of which is constructed with a particular technology and which performs a specific role in the computer system. These memory subsystems range from cache memory to DVD drives.

a) Explain why a computer cannot use a single memory technology and why a memory hierarchy is necessary. Your answer should discuss the role and characteristics (i.e., performance) of each of the individual memory technologies you would expect to find in a high performance general-purpose digital computer. **(15 marks)**

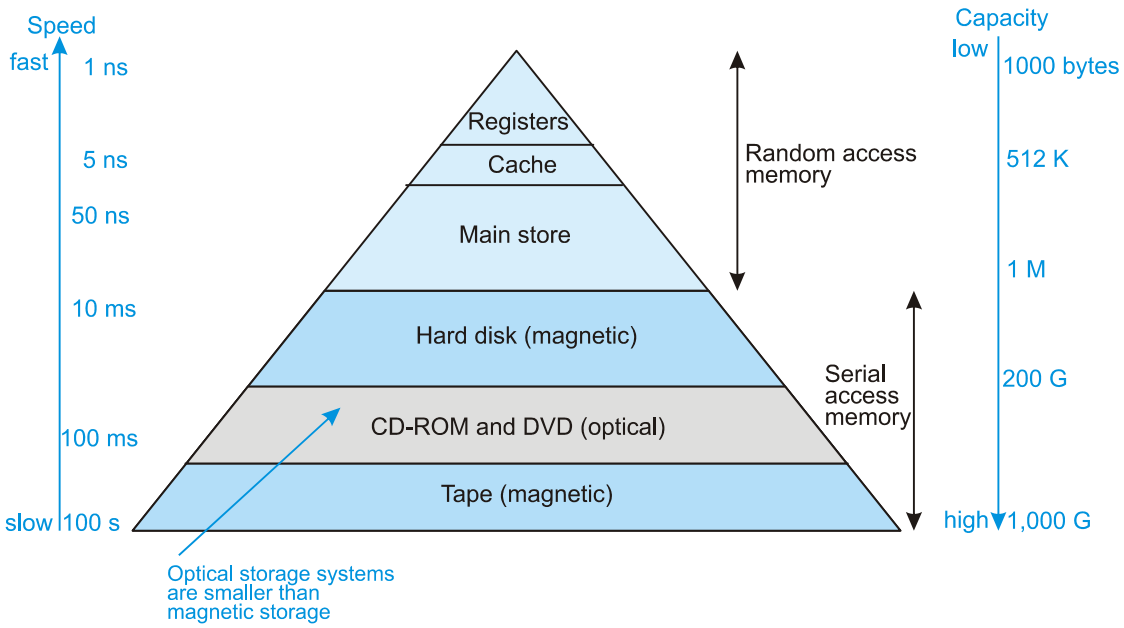
b) Briefly describe the technologies used to implement each level in a memory hierarchy. **(15 marks)**

NOTE: Part *a)* is concerned with the *performance* of the memory subsystems, whereas part *b)* is concerned with the operation and construction of the memory subsystems.

Answer Pointers

(a) A memory hierarchy is required because no single technology provides all the necessary requirements of ideal memory; that is, low-cost, fast access time, non-volatile. By combining various memory technologies, it is possible to create a system with a memory system that has, collectively, the required requirements.

The diagram illustrates the nature of a memory hierarchy.



The components of the memory hierarchy are

a. Cache very fast 1 – 5 ns – high power consumption per bit , volatile. Used only for frequently accessed data in immediate access RAM. Typically 64K to 1 M

b. Main store. Composed of static semiconductor DRAM. Typically 256M to 4Gbytes. Access time 50 ns.Volatile. Used to store working programs data.

c. Flash EPROM. Non-volatile semiconductor memory. Essentially read-mostly (read cycle similar to DRAM but long write cycles). Used for BIOS, MP3 storage and image storage in digital cameras.

D Hard disk. Very large volume storage. (up to 400 Gbyte). Very long access times – 6ms or 10^6 times slower than DRAM. Sequential access.

CD and DVD. Optical storage. Similar to magnetic disk but the media is removable, capacity is smaller (500Mb CD to 4 GB DVD) that magnetic disk and access times much slower. CD/DVD is read-only or can be written one (few times) and writing is slow.

b. I expect student to describe the natures of semiconductor memory, non-volatile flash memory, hard disk or similar magnetic memory, and optical memory. Only basic principles are required.

Examiners' Guidance Notes
Most candidates showed some understanding of memory hierarchy. Reasonable answers were provided on the various components of the hierarchy. Many candidates were unable to distinguish between the performance and the operation of the memory subsystems. Accordingly, answers described almost everything about memory.

Section B

Question 5
5. Describe what the following terms mean:
- *a)* Interpreter
- *b)* Translator
- *c)* Compiler **(3 x 4 marks)**

Answer Pointers
a) Interpreter: A processor that compiles and executes programming language statements one by one in an interleaved manner. If an error is encountered execution of the program stops. Once the error is corrected, the program has to be interpreted again from the start. Interpreters do not produce object code, so the source code is required in order to run the program. Examples of interpreted programming languages can be given here eg BASIC.

b) Translator: A device that changes a sentence from one language to another without change of meaning. Examples of translators can be given here such as a compiler.

c) Compiler: A program that translates between programming languages and produces object code. The compiler translates a whole program, only when it has compiled without errors can the program be executed. Example of programming languages which use compilers can be given here eg PASCAL.

Examiners' Guidance Notes
This question was fairly popular. It was designed to determine whether candidates understood the difference between each of the terms. On the whole it was answered quite well but some students achieved poor marks because their answers were too brief. This indicates poor examination technique. Fuller answers with possible comparisons between the terms were awarded maximum marks.

Question 6
6. *a)* Draw a truth table for the following function: AB + CD
b) Implement the function using NAND gates only **(2 x 6 marks)**

Answer Pointers
a) Truth table for the function AB + CD should have 16 entries because it has four terms ($2^4 = 16$).

b) A double compliment of AB+CD should yield ((AB)'(CD)')', which can be implemented with 3NAND gates.

Examiners' Guidance Notes
This question was slightly less popular than question 5 nevertheless it was answered by approximately half of the total number of candidates. Most candidates were able to gain full marks for drawing the truth table. This part was quite straight forward. Not as many were able to implement the function using NAND gates only and gave confused answers. For those who knew how to implement the function correctly, this was an easy 12 marks to gain.

Question 7

7. Perform one-bit right circular rotation on the following byte patterns, giving your answers in hexadecimal:

a) CD
b) BE
c) 37 **(3 x 4 marks)**

Answer Pointers

a) CD: 11001101 (Binary), right shift is 11100110 which is E6 in hexadecimal.
b) BE: 10111110 (Binary), right shift is 01011111 which is 5F in hexadecimal.
c) 37: 00110111 (Binary), right shift is 10011011 which is 9B in hexadecimal.

Examiners' Guidance Notes
Again a straightforward question but the least popular with the candidates. Most candidates lost marks because they only produced the binary representation and either did not do the right shift or did it incorrectly.

A few candidates answered the question accurately and gained an easy 12 marks.

Question 8

8. In programming languages, what is the difference between:

a) Global variable and local variable
b) Boolean and Floating-point data types **(3 x 4 marks)**

Answer Pointers

a) Global variable and local variable: the answer should focus on the difference in scope between the two types of variables. The answer should describe instances where each type of variable would be suitable and also use an example to illustrate the points being made. Global variable are available for use anywhere in the program whereas local variables can only be used by the module/procedure/function in which they have been defined. Local variables are used in control constructs such as for loops. Global variables can be used for declaring constants.

b) Boolean and Floating-point data types: Boolean data types record true / false logic, they can be represented using a single bit where 0 is false, 1 is true. Boolean data types can be used for recording the result of a condition. For example 'If overdue then ….' . In this statement the statements will only be executed if the overdue condition is true. Floating-point data types are used for storing real numbers. The number consists of a mantissa and an exponent. For example $3.14 = 0.314*10^1$
Boolean data types take up less storage than Floating point data types.
Floating point data types can be used for calculations, Boolean data types cannot.

Examiners' Guidance Notes
Candidates performed better on a) than they did for b). The answers for section b lacked examples and comparisons between Boolean and Floating-point data types.

Question 9

9. Describe, with illustrations, THREE different network topologies. Outline one advantage and one disadvantage of each topology described. **(12 marks)**

Answer Pointers
Star topology – nodes communicate with the central computer. Typical advantage is that if a node fails this does not affect others. One of the disadvantages is that if the central node fails the network is unusable.

Mesh topology – every computer linked to every other computer. Advantage is that if a node fails the effect is minimised since the rest can carry on communicating and another path can be used. One disadvantage is the cost of the network is high.

Bus – all computers connected to common bus, only one computer is allowed to transmit onto the bus, all others receive. Advantage is that the cost of adding a new node is reasonable. A disadvantage is that if the cable/backbone fails the network is unusable.

Ring – bits of a message being transmitted round the ring without waiting for the rest of the message to which they belong. Advantage is simultaneous access, disadvantage is that failure of the ring is fatal.

Other advantages and disadvantages are also possible for each of the above.

Examiners' Guidance Notes
This was the most popular question by far and most students scored highly for this question. In a very few cases the students lost marks because they were confused between star and ring topologies.

It was disappointing to see one or two students who did not read the question properly. The question clearly states '… with illustrations …'., yet they did not draw the diagrams for each of the topologies so marks were lost as a result. Secondly some students (not many) listed either the advantages or the disadvantages but not both so once again easy marks were lost.

Question 10

10. State the functions of the following:
a) ALU
b) Program Counter
c) Instruction Register **(3 x 4 marks)**

Answer Pointers
a) ALU – part of cpu where all arithmetical and logical operations are performed. Eg add, complement, and, or, exclusive or, shift, test
b) Program Counter – keeps address of next instruction to be executed. Initially loaded with the address in memory of the location of the first instruction of the program, and when next instruction is fetched, the counter is incremented to point to the next location and so on.
c) Instruction register – holds current instruction being executed.

Examiner's Guidance Notes

This was pure book work. This question was answered reasonably well, however some students achieved poor marks because their answers were too brief. In other cases students got their answers mixed up for Program Counter and Instruction Register. This indicates poor examination technique. Fuller answers which included diagrams/examples were awarded maximum marks. In other cases students got their answers mixed up for Program Counter and Instruction Register.

Question 11

11. Convert the following numbers to their decimal value:

a) 1100.1110

b) BAE

c) FE+1D **(3 x 4 marks)**

Answer Pointers

a) $1100.1110 = 1* 2^3 + 1*2^2 + 0*2^1 +0*2^0.1*2^{-1}+1*2^{-2}+1*2^{-3}+0*2^{-4}$

$= 8 + 4 + 0 + 0. ½+1/4+1/8+0$

$= 12.875$

b) $BAE = E*16^0 + A * 16^1 + B * 16^2$

$= 14 * 1 + 10 * 16 + 11 * 256$

$= 14 + 160 + 2816$

$= 2990$

c) $FE + 1D = 14 * 16^0 + 15 * 16^{1+} D* 16^0 + 1 * 16^1$

$= 14 + 240 + 13 + 16$

$= 254 + 29$

$= 283$

Examiner's Comments

Again a straight forward question but not very popular with the candidates. Those who did attempt the question managed to score highly with a few exceptions.

Question 12

12. In your own words, explain what is meant by the following terms:

a) Artificial Intelligence

b) Expert Systems

c) Robot **(3 x 4 marks)**

Answer Pointers

a) Artificial Intelligence – Answer should give a one or two paragraph description of AI. One definition of AI is ' a branch of computer science concerned with making computers behave like humans. There are many branches of AI such as neural networks, expert systems, fuzzy logic and so on'.

b) Expert Systems – answer should define what an expert system is, say what type of applications it can be used for, give some examples of expert systems, and explain the AI techniques used in expert systems.

c) Robotics – answer should explain in general how a robot works, its advantages, limitations and give examples of areas where robots have been used

Examiner's Guidance Notes
Section a) was poorly attempted. Most students not recognise the difference between AI and expert systems so their answer for part a) and b) was virtually the same. Part c) was answered quite well.