BCS HIGHER EDUCATION QUALIFICIATIONS
BCS Level 4 Certificate in IT

April 2010

EXAMINERS' REPORT

Computer and Network Technology

This report is for Sections A and B of the paper. It consists of two sections:

1. Overall comments
2. Detailed questions' report

1. Overall comments

As in previous sittings, many candidates did not follow instructions and accordingly did not encircle the question numbers at 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 impress this rule on their candidates as omission of vital information causes problems during marking. The level of English was bad in some centres, hence the inability for candidates to express themselves clearly.

While marking scripts from various centres, it was felt that candidates did not seem to have studied the various topics well. Course providers must endeavour to explore this syllabus in depth. Computer and Network Technology is a core aspect of an IT Professional. Gaps in knowledge with regards to this area can pose serious problems in understanding more advanced and specialist IT concepts. If in doubt, course providers must seek clarification from the BCS examinations office.

It was also noted that many candidates did not attempt the correct number of questions. Consequently, they were not able to score enough marks to achieve a pass grade.

2. Detailed questions' report

It is 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. It is also worth mentioning that candidates seemed to ignore the amount of marks allocated to questions in section A and B. When writing answers, candidates must carefully keep in mind that questions 1, 2, 3 and 4 each carries 30 marks. Accordingly, candidates must write sufficiently in-depth answers to attract these marks. Short answers which lack depth did not enable candidates to score well in section A.

Question 1

A1. a) The *stored program* von Neumann digital computer executes instructions in a *two-phase fetch/execute* mode. Explain what we mean by the expressions ***stored program*** and ***fetch/execute*** used in this statement.

(6 marks)

b) With the aid of a diagram describe the structure of a digital computer (in terms of registers, buses, and functional units) and explain how instructions are executed.

(12 marks)

c) Describe the machine (assembly language) instructions that are executed by a typical computer. The computer may be a commercially available machine or a conceptual machine that you have studied on your course. Illustrate your answer with several different types (or class) of assembly language instruction to demonstrate the range of operations these instructions may carry out.

(12 marks)

Answer points

a) A stored program computer is a machine in which the program and data share the same memory. The notion of a stored program computer dates back to the mid 1940s and is associated with von Neumann (von Neumann and stored program computer are interchangeable). Most computers we use today are stored program machines. However, because data and instructions are cached in separate memories in high-performance microprocessors, some people could say that today's computers are no longer truly stored program machines.

Because data and program (instructions) reside in the same memory it is necessary for a computer first to read an instruction from memory, to decode it, and then to read and data it needs from memory (or write a result back to memory). Thus, you can say that each instruction has a fetch phase and an execute phase requiring two accesses to memory per instruction. This mode of operation gives rise to the so-called *von Neumann bottleneck* because of the extensive memory-CPU address and data traffic.

To sum up: In a stored program computer, program and data co-exist in the same memory and it is even possible for a program to modify itself during execution. Each instruction is read (the fetch phase) and then internally decoded and executed. The execution phase may require one or more additional memory accesses to data.

Note – modern computers with multiple cores, multithreading, parallel execution and cache memories are not von Neumann machines in the strict sense of the term. However, the stored-program fetch/execute concept gives you a conceptual idea of how computers operate.

b) The diagram below describes the structure of a CPU. Different candidates may describe the processor in different ways. However, the key feature is the **program counter** that contains the address of the next instruction to be executed. The program counter is connected to the program and data memory. The processor reads the current instruction and transfers it (via the memory buffer register) to the instruction register.

A CPU has at least one data register and buses between the data register and memory buffer register to the ALU where operations are carried out. The output of the ALU is connected to both the data register (accumulator) and the memory buffer register (in order to write results into memory).

Registers are connected to other registers and functional units like the ALU by buses that transfer data from point to point under the control of the control unit.

Using the diagram on the next page we can explain the fetching and execution of an instruction (strictly speaking, the question does not ask students to define the fetch phase and a student should not be penalized for omitting this.

1. The contents of the program counter (i.e., the pointer to the next instruction in memory) are copied to the memory address register. That is (in RTL) $[MAR] \leftarrow [PC]$

2. The contents of the program counter are also sent to the incrementer in order to point to the next instruction in sequence, and the incremented value loaded in the PC.

That is, [PC] ← [PC] + 1 NOTE – in reality the increment is 2 to 10 (or more) because instructions can take a variable number of bytes in computers like the Pentium or 68K

3. The memory address register, MAR, currently contains the address of the next instruction to be executed. The memory uses this address and looks up that actual instruction op-code which is deposited in the memory buffer register, MBR. The MBR is sometimes called the memory data register. The read operation can be described as, [MBR] ← [[MAR]]

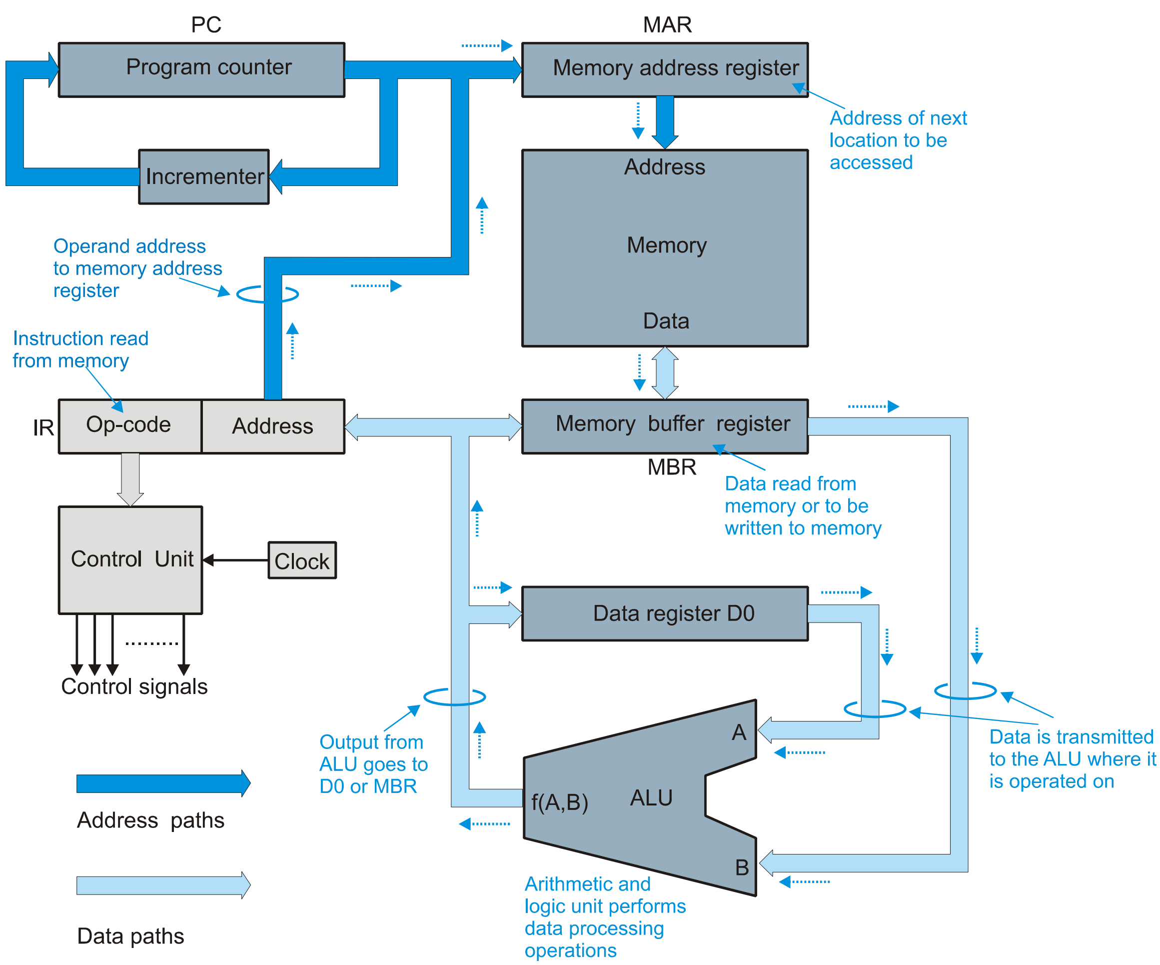
4. The instruction copied into the MBR is transferred to the instruction register, IR. The instruction register holds instructions while they are being executed. [IRC] ← [MBR]

5. The processor's control unit takes the instruction op-code bits and uses them to generate the signals (clock pulses, bus control, ALU and memory control).

6. Typically, an instruction might be, say, ADD P,D1 where P is a memory location and D1 a register. The computer now routes the address field of the IR to the memory address register and then performs a second read – this time to get the operand P. The computer then copies both P and the value of D0 to the ALU where P and D0 are address.

7. Finally, the output of the ALU transferred to the data register. However, if the instruction is ADD D1,P, the result would be sent to the MBR and a third memory access take place.

Candidates should be able to give a coherent description of the sequence of actions that take place in a computer as an instruction is fetched from memory and executed,



c) This is a fairly open-ended question. I am not looking for a specific answer. Students should be given appropriate credit for any answer that displays a coherent understanding of the nature of an instruction set. NOTE that the intention of the question is to describe instruction types and not instruction formats (i.e., the question does not ask about instruction syntax or addressing modes). However, if a student misinterprets the question and covers, for example, addressing modes, the marking should be carried out sensitively – as long as the answer is coherent and well argued.

Typical instruction classes are:
Data movement:
MOVE register to register, or register to memory, or memory to register, or memory to memory. Such instructions are provides by all computers. For example; MOVE D0,D2. Some computers use LOAD and STORE instructions rather than a generic move (e.g. LOAD r1,(r2) or STORE d3,(d7)) in which data is moved to a register (load) or moved from a register (store).

Some computers have variations on move such as SWAP (or EXG) that swap/exchange pairs of registers. Some computers have instructions that operate on fields of a register (or memory location) to allow to access data that is not of the standard word length.

Arithmetic
These are the core instructions required by all computers. All computers include ADD and SUBTRACT instructions that from the sum or difference of two values; there are many formats depending on whether the computers is a 2 or 3 address machine (e.g. ADD D1,D2 or ADD r1,r2,r3).
Some computers have multiplication and division instructions (often in two versions for signed and unsigned arithmetic). More complex arithmetic may exist in computers with floating-point instruction sets (e.g. sines and cosines).
NOTE Some regard test and compare instructions as arithmetic operations – such as CMP #4,D6 (compare D6 with 6).

Logic
Logic instructions are similar to arithmetic instructions and include the operations AND, OR, NOT, and EOR. All computers include a set of logical operations. Logic operations are required to operate on the individual elements of a word to set, clear, or toggle bits.

Shift
Shift operations act on a single word and move the bits one or more paces left or right. However, there are several classed of shift depending on whether the data is regarded as logical, or arithmetic. Typical operations are LSL #2,D4 (logically shift left the contents of register D4 by two places) or ROR #6,D7 rotate right by 6 places the contents of D7.
Shift operations are used in three applications: arithmetic (shifting is multiplying or dividing by 2), data processing (to access a field in a word), or multiprecision arithmetic.

Flow control
These are the instructions that control the sequencing of instructions; for example subroutine calls and returns (BSR/JSR and RTS).

Conditional branch instruction allow the computer to change the sequence of execution depending on the outcome of an operation; for example,
BEQ Next (branch to the line labelled 'Next' if the last result was zero). These instructions are needed to implement if..then..else operations and while..for/until loops.

Consider the tiny fragment of code

```
      MOVE #10,D0 data move
      MOVE #0,D1  data move
Loop  ADD  D0,D1  arithmetic
      SUB  #1,D0  arithmetic
```

```
BNE  Loop   flow control
```

Question 2

Operating systems have come a long way from the primitive punched-card-based systems of the 1960s to today's operating systems like Windows 7™ or Mac OS X™.

a) Give a brief account of the history of the operating system and indicate some of the key developments in its history.

(6 marks)

b) Explain the role and functionality of the operating system in a modern high-performance personal computer or work station.

(18 marks)

c) Briefly state how you think that operating systems might develop in the next decade?

(6 marks)

Answer points

a) The history of the computer is as much the history of the operating system as the history of hardware. The first computers did not have an operating system. Programs were either hardwired (i.e., could not be changed) or were loaded manually. There was no automatic control of peripherals or filing systems (you read data from specific locations in memory).

The first operating systems were single-user operating systems that allowed you to load a program (typically on paper tape) and to run it.
Later, Job Control Languages, JCLs, were developed to allow the computer operator to load a batch of programs (usually on card). The JCL could also control I/O devices and storage. A JCL is, essentially, a batch operating system.

Because computers were expensive in the 1960s and 70s, timesharing operating systems were developed that allowed people to interact remotely with computers using dumb terminals.

In the 1960s multitasking operating systems were developed that allows several programs to run simultaneously. Equally, virtual memory systems were developed that allowed the automatic management of disk drives (transferring data between high speed RAM and disk).

The Personal Computer led to the design of windows-based (GUI) operating systems that provides a visual front-end to the user and automatically managed program execution, memory management, and communication (via the Internet).

Real-time operating systems have been developed for control applications – these ensure that time-critical operations are catered for (e.g. in control systems).

b) The role of an operating system is to free the computer user (operator) from all worries and concerns about the control f a computer and all its resources. The operating system converts a computer with real (actual) hardware and software into an abstract or virtual machine that carries out the operations required by the user – without the user having to know anything of the underlying hardware.

The traditional role of the operating system is memory management. This manages all storages devices in the memory hierarchy from DRAM main store to disk drives to optical storage and even UBS memory stick plug-ins.

The modern graphical operating system is able to accept input from the user via keyboards, mouse movements, and touch screens. Mouse movements and clicks (via, for example, pull-down menus) etc are analyzed by the operating system and the appropriate actions implemented.

For example, if you wish to copy a file from a hard disk to a DVD you simply select the file on the hard disk and drag it to the icon for a DVD. The operating system detects the intent (move file) and then performs all the operations required. This may require the reformatting of data between differing media, error handling, and data routing. All this is transparent to the user.

Modern operating systems have integrated web browser technologies into the operating system in an attempt to make the computer itself look like an extension of the internet will no fundamental differences between local files and applications and those stored remotely on the Internet. The operating system also handles all communications protocols and I/O devices such as printers and scanners.

c) It is impossible to anticipate fundamental changes – by their very nature they are revolutionary. However, since the introduction of the graphical operating system two decades ago, there has been no revolutionary change in operating stems. Change has been evolutionary and not revolutionary. Change has incorporated the latest technical changes into the mainstream OS; for example, WiFi cards once required a lot of user intervention and setting up – nowadays that is handled automatically by the operating system and user intervention is often carried out by an operating-system-driven dialogue.

Operating systems of the future will cater better for new improved input such as touch screens and include better (and more predictive) interfaces.

In general there is a continued convergence in digital technology with functions normally carried out by one device implemented in another – a good example is voice communications over the Internet (exemplified by Skype). This will probably continue and the PC will take on more of the functions of a media control system (streaming video and music) and handling voice communications.

More and more functionality is being built into operating systems. For example, Internet security is currently largely handled by third-party add-on software (mail filters and virus detectors). It would be reasonable to see such malware detectors built into operating systems themselves in the future (the limitation here is political and economic because the incorporation of features in an operating system could be seen as anti-competitive). More effort may be made to ensure that operating systems are self repairing when they encounter either malware or hardware failures.

Operating systems may be affected by the move to cloud computing and distributed software/data. For example, it would be possible to move from the 'buy software' model to the 'pay-by-the-drink' model where most software is on remote computers and you pay to access a word processor when you need it.

Question 3

a) With the aid of one or more diagrams, explain how a hard disk drive operates.

(12 marks)

b) What are the fundamental characteristics (operating parameters) of a modern high capacity hard disk drive in a PC or work station?

(6 marks)

c) All modern hard disk drives have an internal cache memory. Describe the purpose of this cache memory and briefly explain how it operates.

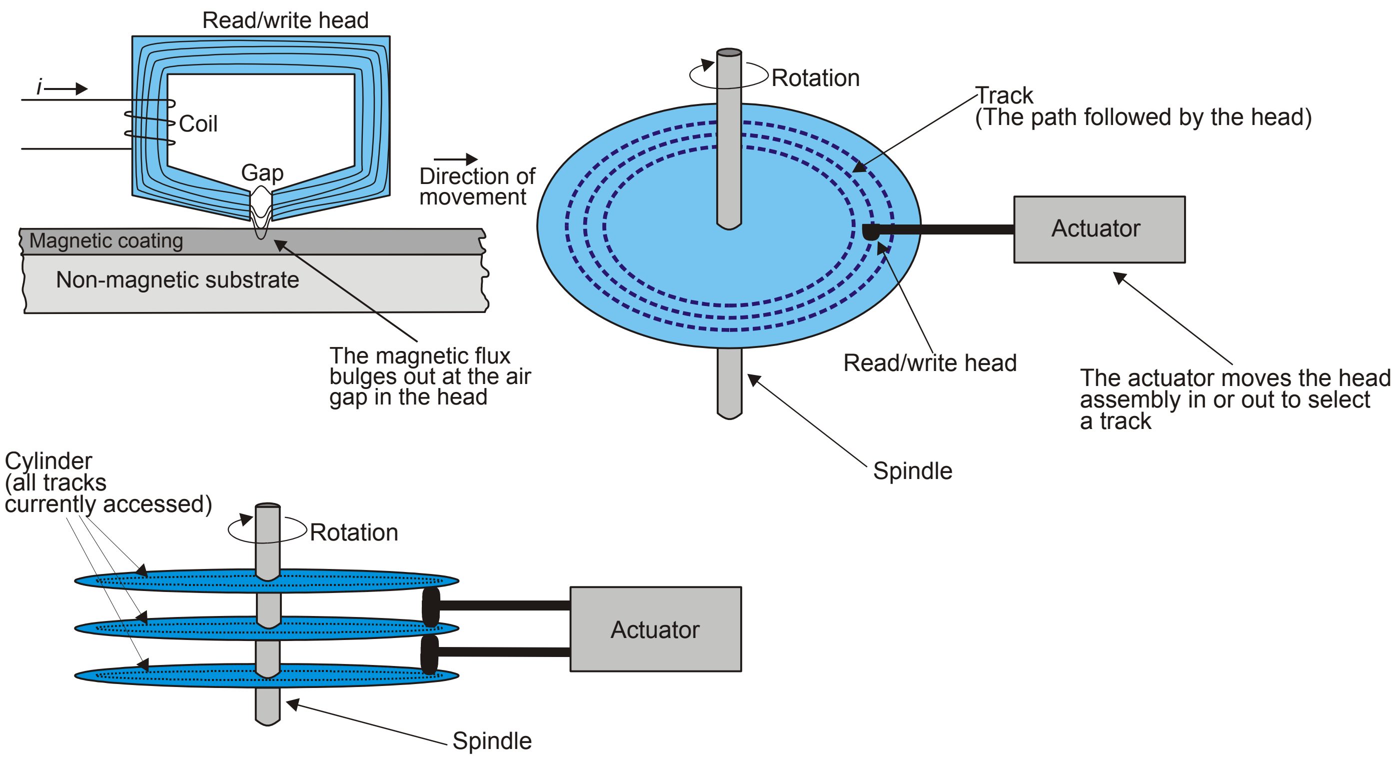
(6 marks)

d) Optical storage mechanisms such as CD, DVD, and Blu-ray are also used to store data. In what way do the operational characteristics of optical storage media differ from magnetic media?

(6 marks)

Answer points

a) The following three diagrams explain how a tape head writes data onto a track by creating a local magnetic field that penetrates the surface of a platter covered with magnetic material. As the disk rotes, the path of the head over the surface is a circle or track and the bits are recorded along the track. A track represents too much data to read or write at once, so a track is divided into smaller slices called sectors.



The head is on an arm that can be stepped in or out to select one of many different tracks. Finally, a real disk has multiple platters each with their own heads so that if track m is selected on one disk track m is also selected on all other disks.

To write data on a disk, the head is stepped to the correct track. The data is then read until the start of a track is located and then sector are read until the correct sector is found (disks are sequential access devices because you can step to a given track but have to wait for a given sector to rotate under the head). Then the data can be written to the sector.

The same process is repeated when reading a disk, except that the sector is read not written.

b) A disk is described by its physical size or form factor (typically 3.5 inch for PCs and workstations, and 2.5 inch for laptops, 1.8 inch for iPods), its data capacity (usually over 100 Gb today with a maximum of 2TB in 2010), and it's rotation speed – 7,200 rpm for most PCs, 15,000 rpm for very high performance systems, and 5,400 rpm for small disks in laptops.

Data transfer rate is more complex to measure as data is read from the surface of the disk at about 70 mbytes/s and then transferred from the buffer to the processor at 3 Gbits/s.

The seek time of a disk is the time to step to a required track. This is complex to measure but is normally in the range 2ms to 9 ms.

Other parameters include the areal density (bits per square inch of surface), track spacing, and sensitivity to shock (important in mobile applications with a tolerance of 350G for rotating disks and 1000G for stationary disks).

Finally, disks are characterized by their interface to the host system – this can be SCSI, ATA, or SATA. The serial SATA interface is the current standard because of its high-performance, and low-cost.

c) Most computer architecture texts cover cache memories in computers and virtual memory. However, today's high-performance hard disks invariably include internal cache memory. Because disks are relatively slow in comparison with semiconductor memory, data from a hard disk is internally cached in semiconductor memory buffers (this is performed automatically and is invisible to the user – that is, you have no control over how disk caches operate).

By caching data, it is not necessary to re-read the disk if data is access multiple times. Moreover, data can be written rapidly to the cache and then later transferred to the disk.

Typical cache sizes are 2 to 32 Mbytes with 16 Mbytes being common on 2010.

d) Optical storage systems are similar to disk storage systems. Both store data on a rotating flat medium. Optical media are generally not protected from the elements which means they are subject to the effects of dust and bad handling. However, optical media are used to transport data and archive it (ranging from 600 Mbytes (CD) to 4.7 GB (DVD) and over 50 Gb for Blu-ray. All three systems use a single spiral track (not concentric tracks as in a magnetic disk) to store data in the form of indentations in the surface. A laser beam detects the indentations because the path length of light falling of the surface and top of an indentation differed by a half wavelength of light. As time has passed, engineering improvements and laser design (higher laser frequencies from infrared to blue) has allowed a greater bit density.

The optical storage system is intrinsically read-only and the first medium was the CD designed for audio use. Optical storage allows the mass production of low-cost music, video and data (programs). Writable optical storage has been developed by using thermal or magnetic properties of matter to change the surface of a track (modifying reflectivity or polarization properties). However, writing to an optical storage medium is relatively slow – orders of magnitude slower than hard disk.

Both magnetic disks and optical disks have increased in capacity over the last decade. On the other hand, as electro-mechanical rotating devices, both has seen relatively little improvement in performance – access tome and data transfer times.

In the short term, optical storage will remain largely as a data archiving medium.

Question 4

An alarm system has four digitals inputs, p, q, r, s and one output, f. The output is asserted (set to a logical 1) if the inputs indicate a dangerous situation.

The alarm inputs are:

Temperature sensor 1	p
Temperature sensor 2	q
Motion detector	r
Sound detector	s

An alarm is sounded if any of the following conditions are true:

- Both temperature sensors are activated
- The motion detector and sound detector are both activated
- One temperature sensor and either the motion detector the sound sensor is activated
- If ever the sound detector and temperature sensor 2 are activated then the alarm is not asserted irrespective of any other input conditions (i.e., this state overrides all others).

a) Construct a truth table with four inputs, p, q, r, s, representing the four sensor states and alarm output f.

(8 marks)

b) Using Boolean algebra write down an expression (unsimplified) for the alarm output f.

(7 marks)

c) By means of Boolean algebra or a Karnaugh map write down a simplified Boolean expression for the alarm output f.

(7 marks)

d) Using AND, OR, and NOT gates design a circuit to generate the alarm signal, f, from the sensor inputs.

(8 marks)

Answer points

a) Truth table

Temp sens 1	Temp sens 2	Motion det	Sound det	Alarm	Note
p	q	r	s	F	
0	0	0	0	0	
0	0	0	1	0	
0	0	1	0	0	
0	0	1	1	1	Motion and sound
0	1	0	0	0	
0	1	0	1	0	Temp and sound BUT excluded
0	1	1	0	1	Temp and motion
0	1	1	1	0	Motion and sound BUT excluded
1	0	0	0	0	
1	0	0	1	1	Temp and sound
1	0	1	0	1	Temp and motion
1	0	1	1	1	Motion and sound
1	1	0	0	1	Both temp
1	1	0	1	0	Both temp BUT excluded
1	1	1	0	1	Both temp
1	1	1	1	0	Both temp BUT excluded

The rules are: p+q rule 1

r+s rule 2
(r⊕q).(r+s) rule 3
s.q = 1 THEN F = 0 rule 4

From the table, we can write

NOTE – some students may be confused by rule 3 (*One temperature sensor and either the motion detector or the sound sensor is activated*) This does mean ONLY one temperature sensor. However, if students use the case that includes both temperature sensors, the answer was accepted.

b)

$F = \overline{p}.\overline{q}.r.s + \overline{p}.q.r.\overline{s} + p.\overline{q}.\overline{r}.s + p.\overline{q}.r.\overline{s} + p.\overline{q}.r.s + p.q.\overline{r}.\overline{s} + p.q.r.\overline{s}$

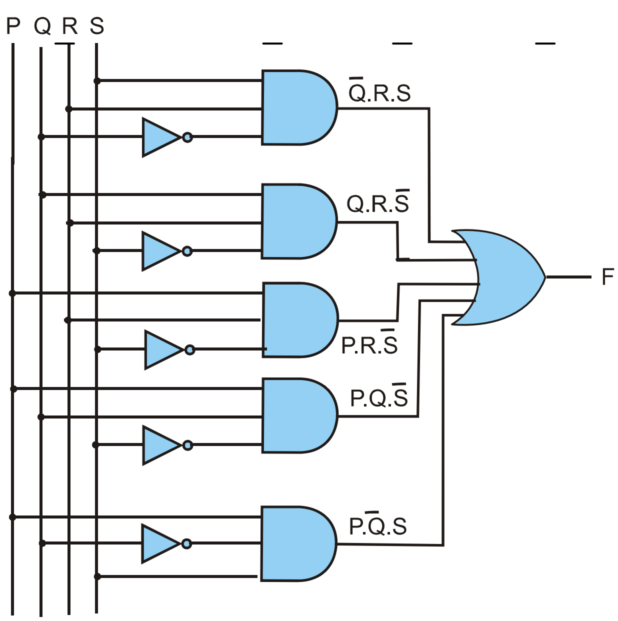
c)

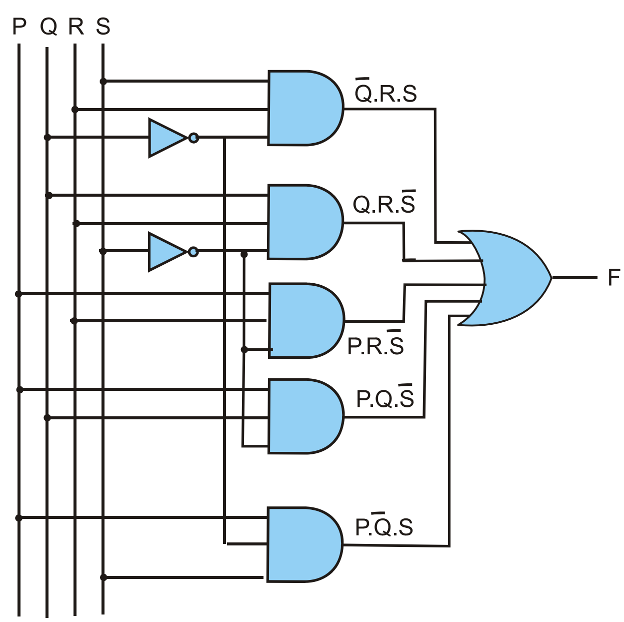
rs \ pq	00	01	11	10
00			1	
01				1
11	1			1
10		1	1	1

rs \ pq	00	01	11	10
00			1	
01				1
11	1			1
10		1	1	1

F = $q.r.s + q.r.s + p.r.s + p.q.s + p.\overline{q}.s$

d)





Note: Either one of the diagrams is fine.

Question 5

Communication protocols provide various functions during data transmission.

a) Describe what you understand by communication protocols.

(6 marks)

b) Explain where and how DHCP is used.

(6 marks)

Answer points

a) All communication between devices require that the devices agree on the format of the data. The set of rules defining a format is called a protocol. At the very least, a communications protocol must define rate of transmission. Protocols can also include sophisticated techniques for detecting and recovering from transmission errors and for encoding and decoding

b) DHCP - Dynamic Host Configuration *Protocol,* a protocol for assigning dynamic IP addresses to devices on a network. With dynamic addressing, a device can have a different IP address every time it connects to the network. In some systems, the device's IP address can even change while it is still connected. DHCP also supports a mix of static and dynamic IP addresses.

Question 6

Computer technology has seen lot of development in the recent years. In particular, several semiconductor manufacturers have introduced dual core (or multicore) processors.

a) Explain what a dual-core processor is.

(6 marks)

b) Describe how dual-core processors have improved processing capabilities.

(6 marks)

Answer points

Dual-core refers to a CPU that includes two complete execution cores per physical processor. It has combined two processors and their caches and cache controllers onto a single integrated circuit. Dual-core processors are well-suited for multitasking environments because there are two complete execution cores instead of one, each with an independent interface to the frontside bus. Since each core has its own cache, the operating system has sufficient resources to handle most compute intensive tasks in parallel.

Multi-core is similar to dual-core in that it is an expansion to the dual-core technology which allows for more than two separate processors.

Question 7

a) Briefly describe the concept of network performance monitoring.

(8 marks)

b) Explain why it is important to monitor the performance of a computer network.

(4 marks)

Answer points

a) In network management terms, network monitoring is the phrase used to describe a system that continuously monitors a network and notifies a network administrator though messaging systems (usually e-mail) when a device fails or an outage occurs.

b) Network monitoring is usually performed through the use of software applications and tools. At the most basic level, ping is a type of network monitoring tool. Other commercial software packages may include a network monitoring system that is designed to monitor an entire business or enterprise network. Some applications are used to monitor traffic on your network, such as VoIP monitoring, video stream monitoring, mail server (POP3 server) monitoring, and others. Monitoring is required to detect that the system is operating correctly and to detect any loading issues.

Question 8

a) Differentiate between LAN and WAN.

(6 marks)

b) What is a VLAN?

(6 marks)

Answer points

a) A computer network that spans a relatively small area. Most LANs are confined to a single building or group of buildings. However, one LAN can be connected to other LANs over any distance via telephone lines and radio waves. A system of LANs connected in this way is called a wide-area network (WAN).

b) VLAN - short for virtual LAN, a network of computers that behave as if they are connected to the same wire even though they may actually be physically located on different segments of a LAN. VLANs are configured through software rather than hardware, which makes them extremely flexible. One of the biggest advantages of VLANs is that when a computer is physically moved to another location, it can stay on the same VLAN without any hardware reconfiguration

Question 9

a) Define decimal, binary and hexadecimal number bases. Explain why it is necessary to understand number base conversion.

(6 marks)

b) Convert hexadecimal 4F7B into binary and decimal

(6 marks)

Answer points

a) Refers to the base-16 number system, which consists of 16 unique symbols: the numbers 0 to 9 and the letters A to F. For example, the decimal number 15 is represented as F in the hexadecimal numbering system. The hexadecimal system is useful because it can represent every byte (8 bits) as two consecutive hexadecimal digits. It is easier for humans to read hexadecimal numbers than binary numbers.

To convert a value from hexadecimal to binary, you merely translate each hexadecimal digit into its 4-bit binary equivalent

b) 4F7B hex – Binary 0100 1111 0111 1011 and Decimal 20347.

Question 10

. In the context of computer security, briefly describe the following:

a) Digital certificate

(4 marks)

b) SSL

(4 marks)

c) Firewall

(4 marks)

Answer points

a) Digital certificate - An attachment to an electronic message used for security purposes. The most common use of a digital certificate is to verify that a user sending a message is who he or she claims to be, and to provide the receiver with the means to encode a reply.

b) SSL - Short for Secure Sockets Layer, a protocol developed by Netscape for transmitting private documents via the Internet. SSL uses a cryptographic system that uses two keys to encrypt data − a public key known to everyone and a private or secret key known only to the recipient of the message. Both Netscape Navigator and Internet Explorer support SSL and many web sites use the protocol to obtain confidential user information, such as credit card numbers. By convention, URLs that require an SSL connection start with https: instead of http:.

c) Firewall - system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both. Firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially intranets. All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria.

Question 11

a) Describe and distinguish between the operating principles of a laser printer and an ink-jet printer.

(6 marks)

b) What factors determine whether a user would buy a laser printer or an ink-jet printer?

(6 marks)

Answer points

a) Laser printer - A type of printer that utilizes a laser beam to produce an image on a drum. The light of the laser alters the electrical charge on the drum wherever it hits. The drum is then rolled through a reservoir of toner, which is picked up by the charged portions of the drum. Finally, the toner is transferred to the paper through a combination of heat and pressure. This is also the way copy machines work.

Because an entire page is transmitted to a drum before the toner is applied, laser printers are sometimes called page printers.

Ink-jet printer - A type of printer that works by spraying ionized ink at a sheet of paper. Magnetized plates in the ink's path direct the ink onto the paper in the desired shapes. Ink-jet printers are capable of producing high quality print approaching that produced by laser printers. A typical ink-jet printer provides a resolution of 300 dots per inch, although some newer models offer higher resolutions.

b) Main features include use, quality, purchasing cost and running cost

Question 12

In the context of data security, differentiate between data mirroring and server mirroring.

(12 marks)

Answer points

Data mirroring - The act of copying data from one location to a storage device in real time. Because the data is copied in real time, the information stored from the original location is always an exact copy of the data from the production device. Data mirroring is useful in the speedy recovery of critical data after a disaster. Data mirroring can be implemented locally or offsite at a completely different location.

Server mirroring - Utilizing a backup server that duplicates all the processes and transactions of the primary server. If, for any reason, the primary server fails, the backup server can immediately take its place without any downtime.

Server mirroring is an expensive but effective strategy for achieving fault tolerance. It's expensive because each server must be mirrored by an identical server whose only purpose is to be there in the event of a failure.