THE BCS PROFESSIONAL EXAMINATION
Diploma

April 2000

EXAMINERS’ REPORT

Architecture

Eighteen candidates completed the examination of whom 12 were successful. A detailed description of how each question was answered by the candidates is as follows.

QUESTION ONE

This question was generally answered well. Some candidates failed to define the variables they used in terms of input and output devices. This did not constitute a problem for correct answers but did make it more difficult to deal with answers that were less than completely correct. In some cases circuits were presented without any indication of an expression on which they were based. The NAND and NOR circuits were sometimes constructed by substituting the NAND or NOR gate equivalent of the NOT, AND, OR gates of the circuit of part c on a gate by gate basis. In the worst of these cases redundant gates were not removed.

Answer Pointers

a) Let sensors be A, B and C, error detector be D and state of the alarm be F.

When D = 1 the alarm will never sound.
When D = 0 the alarm will follow the truth table.

A	B	C	D
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

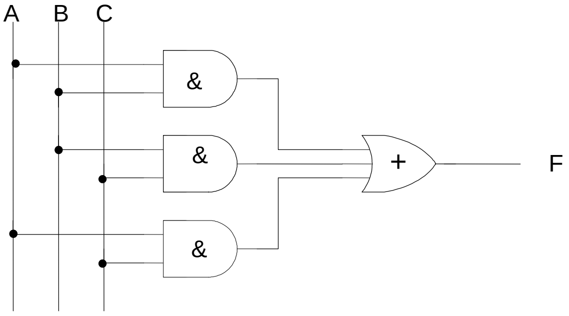
(3 Marks)

b)

	A’B	A’B	AB	AB’
C’	0	0	1	0
C	0	1	1	1

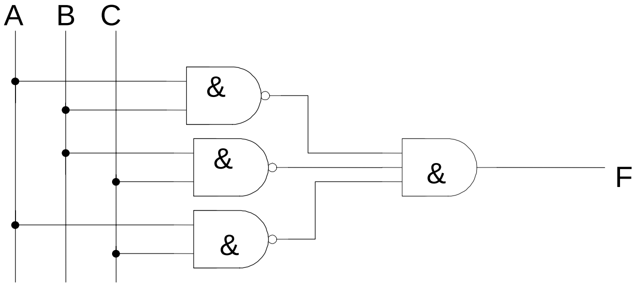
Hence F = A.B + B.C + A.C **(3 Marks)**

6. a)



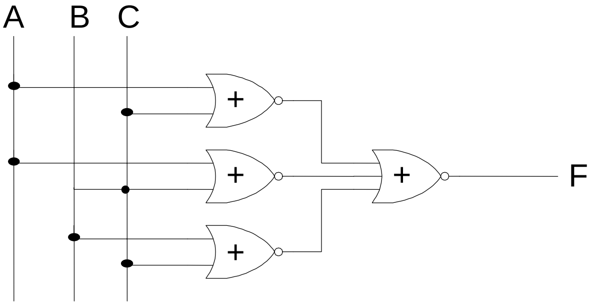
(5 Marks)

d) By deMorgan f = ((A.B)’.(B.C)’.(A.C)’)’



(5 Marks)

e) From drop sets f = ((A+C)’ + (A+B)f + (B+C)’)’



(5 Marks)

f) Propagation delays are the same in all three circuits. The NOT/AND/OR circuit would require two types of IC chip and would be more expensive.

(4 Marks)

QUESTION TWO

This was attempted by 14 candidates. The understanding of the *OSI* 7 *Layer Model* is better than that of recent years. In the option 1A paper. About half the answers showed a poor understanding as to what a *frame* was and what information was needed to accompany the *frame data* for a *Data Link Layer* protocol to work. Similarly, the answers showed rather hazy ideas as to what *flow control* was about.

Answer Pointer

a)

- The OSI 7-Layer model is a template, which lists the services required for effective digital communication and determines how the various services should interact. In particular, in constrains protocols to accept service calls from higher numbered layers and to call on the services of lower numbered layers. A protocol can only interact with its peer layer on remote nodes.
- A standard OSI 7-layer diagram
- 1 mark for a correct description of the function of each layer in the correct order.

(12 Marks)

b)

- Start of frame bit pattern: so that the far end can recognise the start of a frame and can synchronise its clock to the transmitter.
- A mechanism which defines the length of the frame (e.g. a frame terminator sequence)
- A data transparency mechanism (e.g. bit stuffing)

Plus (optionally)

- A sequencing field
- Address field(s)

(6 Marks)

c)

- Flow control is the mechanism whereby a receiver can avoid being overwhelmed by a stream of data from a transmitter by being able to command the transmitter to hold off until such time as the receiver is ready to accept more data.
- There are 2 possible mechanisms:

— Direct control: receiver sends RNR until it is ready then sends RR

— Indirect control: by not acknowledging received data the transmitter soon runs out of ‘credit’ and must stop transmitting until the receiver acknowledges the received data.

(7 Marks)

QUESTION THREE

There was a clear divide between candidates who demonstrated a good understanding of the topic and those seemed to miss the focus of the question. A significant number of answers did not address the software poll and interrupt request.
Few candidates were able to offer a satisfactory list of the factors which determine the interrupt response time in the last part of the question.

Answer Pointer

a) Answers should address most of:

Matching electrical power requirements
Matching speed requirements
Taking delegated control.

(7 Marks)

b) Software poll: Each device interface has a flag it can be set when it requires the attention of the processor. The processor must run a software routine at intervals to check the state of these flags and provide service as required.

Interrupt request: Each device interface is provided with a hardwired connection to the processor. When a device requires attention it signals directly to the processor. The processor checks for interrupt requests between each fetch/execute cycle. Candidates may go on to describe identification of the interrupting device either by means of a software poll or the vectored strategy.

Points for comparison should include most of:

Cost/complexity
Response time
Efficiency in use of processor time
Approach to priority
Suitability for safety critical applications

(12 Marks)

c) This part of the question gives candidates the opportunity to demonstrate their understanding of the sequence of events which occurs when an interrupt request is generated and the approximate timescale for each such event. Answers should address most of:

Finish current fetch/execute cycle
Check priority level
Save machine code environment
Identify device/find service routine

Estimates will depend on the assumptions candidates make.

(6 Marks)

QUESTION FOUR

Some very good answers to this question. A minority of candidates did not take advantage of the don’t care states. The consequences of this were the need to design a circuit of greater complexity with increased potential for error.

Answer Pointers

a) Answers should demonstrate the understanding that when power is first applied to a flip flop is state is indeterminate although it will fall into one of its two stable states after a brief interval. Two precautions could be discussed. The preset and clear inputs might be used to establish an initial value in the counter and/or care could be taken in the design stage to ensure that unused states lead into the main count sequence. The latter requires careful treatment of the don’t care states in the design phase.

b) (I) The state diagram must show a circle of linked states starting and ending with the state 0000 and progressing through the other states in the order set out in the count sequence given in the question.

(II) & (III)

Flip flop A: J = K = 1

Flip flop B

	B’A’	B’A	BA	BA
D’C’	0	1	1	x
D’C	1	x	x	x
DC	0	1	x	x
DC’	x	x	x	x

J = A + C.D’

	B’A’	B’A	BA	BA
D’C’	x	x	0	0
D’C	x	x	x	x
DC	x	x	1	0
DC’	x	x	1	x

K = A..D

Flip flop C

	B’A’	B’A	BA	BA
D’C’	0	0	1	0
D’C	x	x	x	x
DC	x	x	x	x
DC’	x	x	1	x

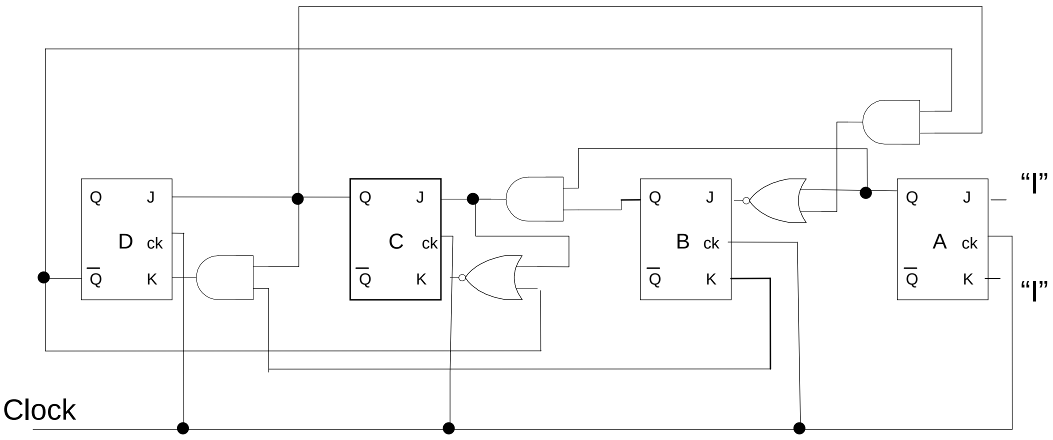
J = A.B

	B’A’	B’A	BA	BA
D’C’	x	x	x	x
D’C	1	x	x	x
DC	0	0	1	0
DC’	x	x	X	x

K = A..B + D’

Flip flop D: J = C K = ABC

(IV)



(20 Marks)

QUESTION FIVE

This was attempted by 11 candidates. Few candidates had any real idea what a *partition* was. Likewise, the understanding of data is held on a disc and the function of a *FAT* was not well understood.

Answer Pointers

a) 1 mark each – 2 marks for partition

- Sector: the minimum addressable amount of disc storage, typically 512 bytes
- Track: the vector of sectors that a head can access without moving
- Cylinder: the vector of tracks which can be accessed without the heads moving
- A Partition: ('Drive' in DOS speak) is a subdivision of a disc usually consisting of a whole number of cylinders. A partition will typically contain a file system, which allows an OS to maintain a collection of files in some form of hierarchical organisation.

(6 Marks)

b)

- A FAT is the logical organisation used by DOS and Windows 3.x to keep track of blocks (clusters) in use by the files in the file system. It also allows DOS to keep track of unused blocks (clusters) and unusable (damaged) blocks (clusters).
- A FAT consists of vector of FAT entries consisting of unsigned 16-bit integers which contain pointers to other FAT entries. The position of an entry in the FAT is an implied reference to a data block (cluster) in the file system.
- The data blocks of a file are defined by a chain of FAT entries; the end of the chain is denoted by a special value (0xFFFF)
- There are two other reserved FAT entry values: 0xFFFE denotes an unusable block (cluster) and 0 an unused (but useable) block (cluster)

(7 Marks)

c)

- A 60MB partition would contain some 12,000 sectors. Because a FAT entry can only address 65532 items, a block (cluster) must consist of 2 sectors (1KB).
- The 4500 byte file will consist of 5 data blocks (clusters)
- A directory will contain an entry consisting of:

— README.DAT
— 3 date/time fields: time of creation, last time updated, last time read
— Attribute byte
— First data block (cluster), *f* say.

- The f^{th} FAT entry will contain a value *g* the g^{th} FAT entry will contain a value *h* the h^{th} FAT entry will contain a value *j*, the j^{th} FAT entry will contain a value *k* and the k^{th} FAT entry will contain the value 0xFFFF denoting the end of the file. The file data blocks are *f, g, h, j and k.*

(12 Marks)

QUESTION SIX

Of the 9 candidates who attempted this question no one achieved more than 17/25. On the whole the examiner feels that candidates should pay more attention to the minor workings of a single CPU.

Answer Pointers

a)

- Instruction fetch: PC → Maddr; PC + = 4; wait for instr in MDR;
- Instruction decode: MDR → IR; set inputs to Q for execution step later.
- Operant fetch: IR $_{\text{(addr)}}$ → b → m → ALU → Maddr; wait for data in MDR;
- Instruction execute: (MDR → IR → b → m → ALU) + (REG → a → ALU) → k → REG

(9 Marks)

b)

The instruction execution time will be Instruction Fetch + Instruction Decode + Operand Fetch + Instruction Execute = 15 + 5 + 15 +5 nS = 40nS.

(8 Marks)

c)

- Cache to reduce effective sore access time. 3nS Cache with 90% hit rate will reduce the effective access time to 4.2nS and the instruction time to 18.4nS. hence about a factor of 2 increase in performance.
- Wider data paths between CPU and store (with cache) to reduce the number of store access cycles. Doubling the number of data lines will halve the effective store access time thus reducing the instruction time to 25nS.
- Faster RAM: changing to 10nS RAM reduces the instruction time to about 30nS; thus increasing performance by a factor of about 1.3
- Faster CPU: doubling the CPU clock rate reduces the instruction execution time to 35nS; thus increasing the performance by a factor of about 1.1

(8 Marks)