Cambridge International AS & A Level

Cambridge International Examinations
Cambridge International Advanced Subsidiary and Advanced Level

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER

COMPUTER SCIENCE **9608/13**

Paper 1 Theory Fundamentals **May/June 2015**

1 hour 30 minutes

Candidates answer on the Question Paper.

No Additional Materials are required.

No calculators allowed.

READ THESE INSTRUCTIONS FIRST

Write your Centre number, candidate number and name on all the work you hand in.
Write in dark blue or black pen.
You may use an HB pencil for any diagrams, graphs or rough working.
Do not use staples, paper clips, glue or correction fluid.
DO **NOT** WRITE IN ANY BARCODES.

Answer **all** questions.
No marks will be awarded for using brand names of software packages or hardware.

At the end of the examination, fasten all your work securely together.
The number of marks is given in brackets [ ] at the end of each question or part question.

The maximum number of marks is 75.

1 **(a)** **(i)** Using two's complement, show how the following denary numbers could be stored in an 8-bit register:

124 | | | | | | | | |

–77 | | | | | | | | |

[2]

(ii) Convert the two numbers in **part (a) (i)** into hexadecimal.

124 .....

–77 .....

[2]

(b) Binary Coded Decimal (BCD) is another way of representing numbers.

(i) Write the number 359 in BCD form.

..... [1]

(ii) Describe a use of BCD number representation.

.....

..... [2]

2 Assemblers translate from assembly language to machine code. Some assemblers translate an assembly language program twice; these are referred to as two-pass assemblers.

The following table shows five activities performed by two-pass assemblers.

Write 1 **or** 2 to indicate whether the activity is carried out during the first pass or during the second pass.

Activity	First pass or second pass
any symbolic address is replaced by an absolute address	
any directives are acted upon	
any symbolic address is added to the symbolic address table	
data items are converted into their binary equivalent	
forward references are resolved	

[5]

3 **(a)** Give the definition of the terms firewall and authentication. Explain how they can [illegible] security of data.

Firewall .....

.....

.....

.....

.....

.....

Authentication .....

.....

.....

.....

[3]

(b) Describe **two** differences between data integrity and data security.

.....

.....

.....

.....

.....

..... [2]

(c) Data integrity is required at the input stage and also during transfer of the data.

(i) State **two** ways of maintaining data integrity at the input stage. Use examples to help explain your answer.

.....

.....

.....

.....

.....

..... [3]

(ii) State **two** ways of maintaining data integrity during data transmission. Use ... help explain your answer.

.....

.....

.....

.....

.....

..... [3]

4 **(a)** There are two types of RAM: dynamic RAM (DRAM) and static RAM (SRAM).

Five statements about DRAM and SRAM are shown below.

Draw a line to link each statement to the appropriate type of RAM.

Statement	Type of RAM
requires data to be refreshed periodically in order to retain the data	
has more complex circuitry	DRAM
does not need to be refreshed as the circuit holds the data as long as the power supply is on	
requires higher power consumption which is significant when used in battery-powered devices	SRAM
used predominantly in cache memory of processors where speed is important	

[5]

(b) Describe **three** differences between RAM and ROM.

.....

.....

.....

.....

.....

..... [3]

(c) DVD-RAM and flash memory are two examples of storage devices.

Describe **two** differences in how they operate.

.....

.....

.....

.....

..... [2]

5 **(a)** Name and describe **three** buses used in the von Neumann model.

Bus 1 .....

Description .....

.....

.....

Bus 2 .....

Description .....

.....

.....

Bus 3 .....

Description .....

.....

.....

[6]

(b) The sequence of operations shows, in register transfer notation, the fetch stage of the fetch-execute cycle.

```
1   MAR ← [PC]
2   PC  ← [PC] + 1
3   MDR ← [[MAR]]
4   CIR ← [MDR]
```

- `[register]` denotes contents of the specified register or memory location
- step 1 above is read as "the contents of the Program Counter are copied to the Memory Address Register"

(i) Describe what is happening at step 2.

.....

..... [1]

(ii) Describe what is happening at step 3.

.....

.....

..... [1]

(iii) Describe what is happening at step 4.

.....

..... [

(c) Describe what happens to the registers when the following instruction is executed:

```
LDD 35
```

.....

.....

.....

..... [2]

(d) **(i)** Explain what is meant by an interrupt.

.....

.....

.....

..... [2]

(ii) Explain the actions of the processor when an interrupt is detected.

.....

.....

.....

.....

.....

.....

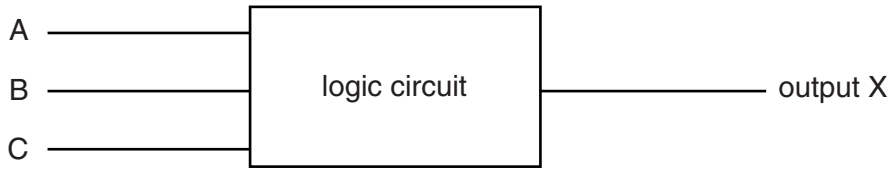
.....

.....

..... [4]

6 **(a)** Three digital sensors A, B and C are used to monitor a process. The outputs from [illegible] are used as the inputs to a logic circuit.

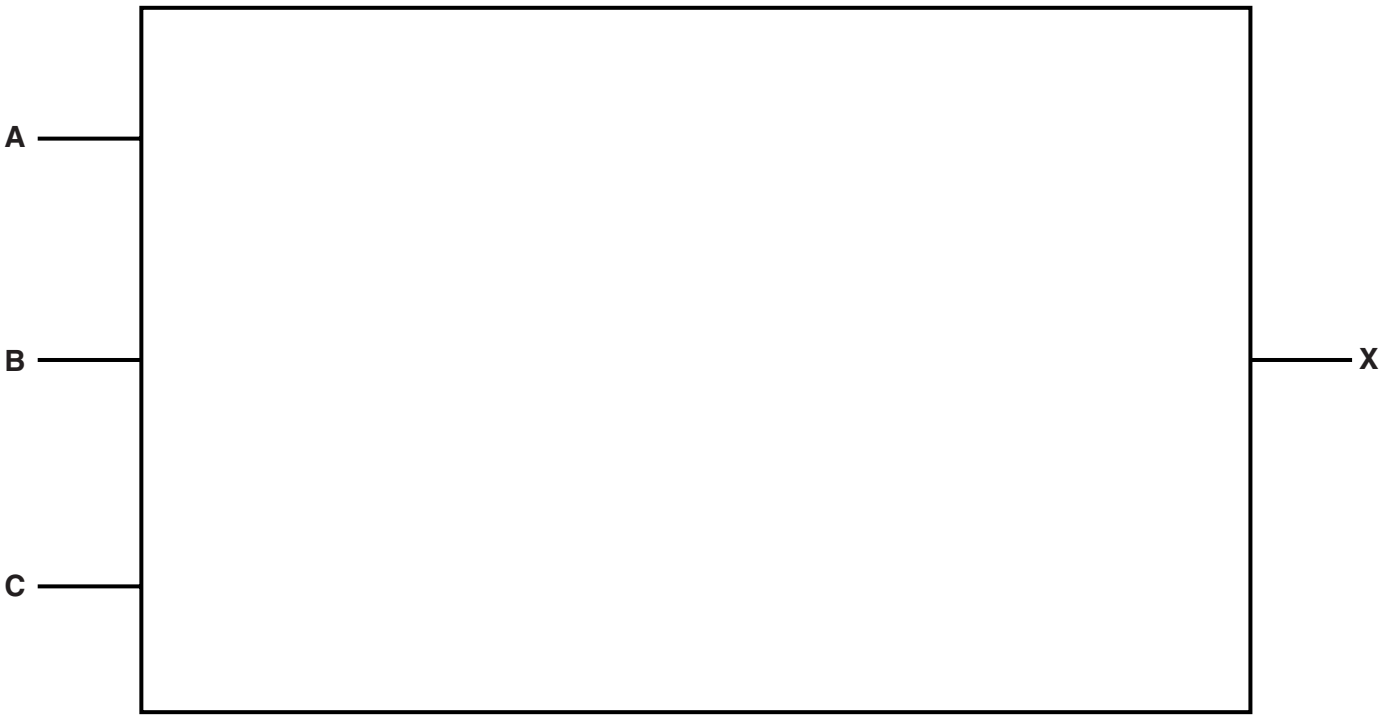
A signal, X, is output from the logic circuit:



Output, X, has a value of 1 if either of the following two conditions occur:

- sensor A outputs the value 1 OR sensor B outputs the value 0
- sensor B outputs the value 1 AND sensor C outputs the value 0

Draw a logic circuit to represent these conditions.



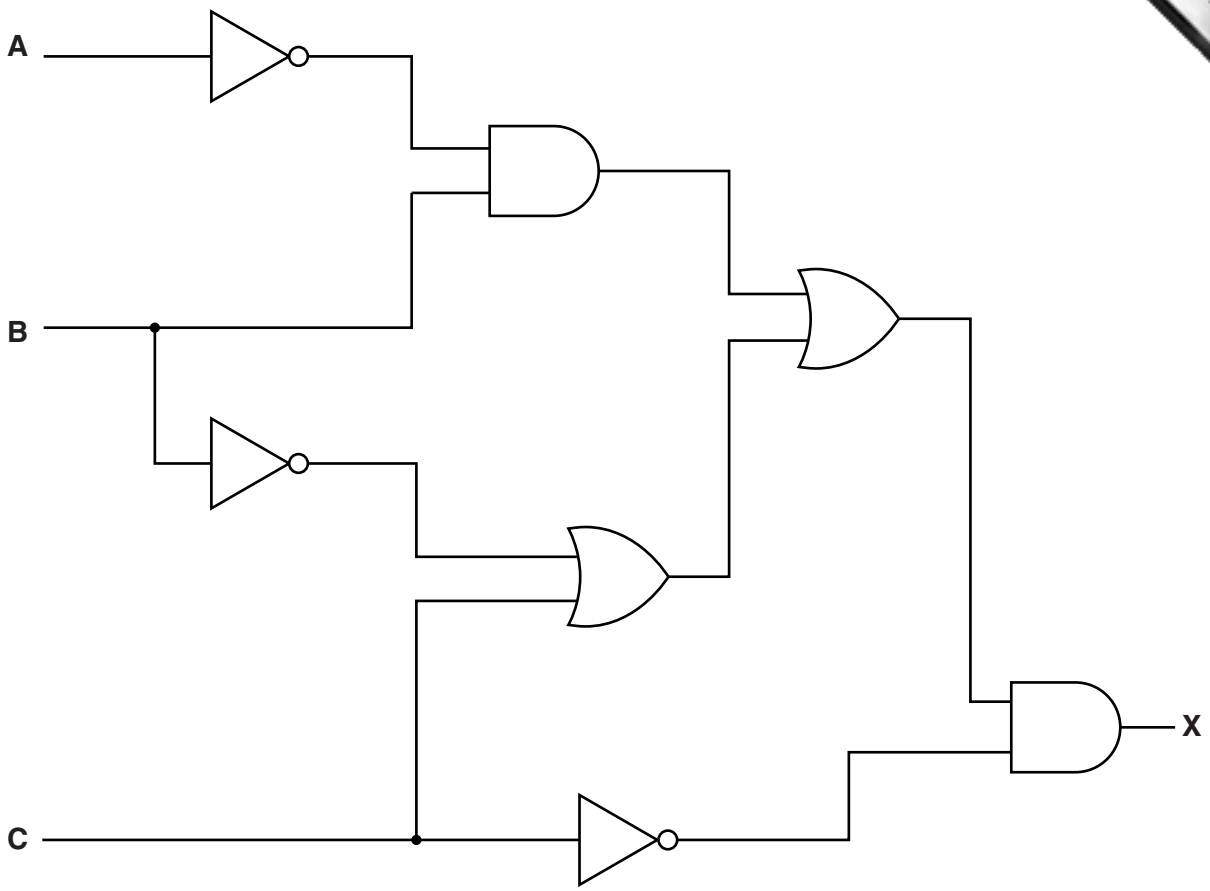
[5]

(b) Complete the truth table for the logic circuit described in **part (a)**.

A	B	C	Working Space	X
0	**0**	**0**		
0	**0**	**1**		
0	**1**	**0**		
0	**1**	**1**		
1	**0**	**0**		
1	**0**	**1**		
1	**1**	**0**		
1	**1**	**1**		

[4]

(c) Write a logic statement that describes the following logic circuit.



.....

.....

.....

..... [3]

Question 7 begins on page 14.

7 The table shows assembly language instructions for a processor which has one general purpose register, the Accumulator (ACC).

Instruction		Explanation
Op code	**Operand**	
LDD	<address>	Direct addressing. Load contents of given address to ACC
STO	<address>	Store the contents of ACC at the given address
LDI	<address>	Indirect addressing. The address to be used is at the given address. Load the contents of this second address to ACC
LDX	<address>	Indexed addressing. Form the address from <address> + the contents of the index register. Copy the contents of this calculated address to ACC
INC	<register>	Add 1 to contents of the register (ACC)
JMP	<address>	Jump to the given address
END		Return control to operating system

The diagram shows the contents of the memory.

Main memory

120	0 0 0 0 1 0 0 1
121	0 1 1 1 0 1 0 1
122	1 0 1 1 0 1 1 0
123	1 1 1 0 0 1 0 0
124	0 1 1 1 1 1 1 1
125	0 0 0 0 0 0 0 1
126	0 1 0 0 0 0 0 1
127	0 1 1 0 1 0 0 1
≀	≀
200	1 0 0 0 1 0 0 0

(a) (i) Show the contents of the Accumulator after execution of the instruction:

LDD 121

Accumulator:								

[1]

(ii) Show the contents of the Accumulator after execution of the instruction:

LDI 124

Accumulator:								

Explain how you arrived at your answer.

.....

.....

.....

..... [3]

(iii) Show the contents of the Accumulator after execution of the instruction:

LDX 120

Index Register:	0	0	0	0	0	1	1	0

Accumulator:								

Explain how you arrived at your answer.

.....

.....

.....

..... [3]

(b) Trace the assembly language program using the trace table.

```
300    LDD    321
301    INC
302    STO    323
303    LDI    307
304    INC
305    STO    322
306    END
307    320
       ↓      ↓
320    49
321    36
322    0
323    0
```

Trace table:

Accumulator	Memory address			
	320	321	322	323
	49	36	0	0

[6]