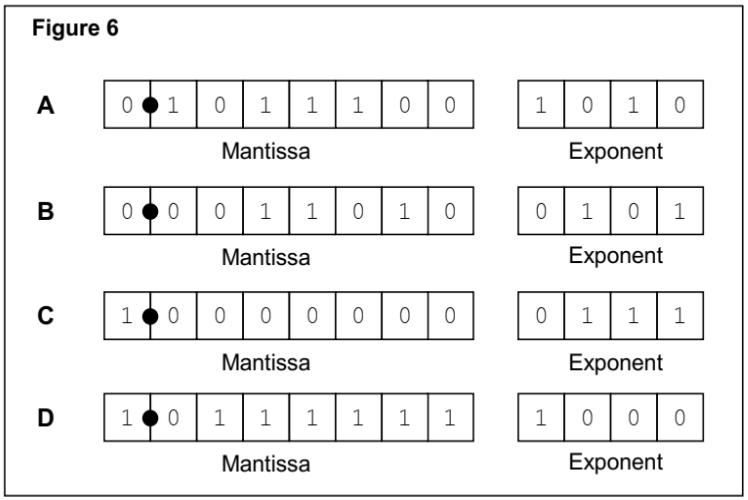
SECTION A

Q1:

A particular computer uses a **normalised** floating point representation with an 8-bit mantissa and a 4-bit exponent, both stored using **two's complement**.

Four bit patterns that are stored in this computer's memory are listed in **Figure 6** and are labelled **A**, **B**, **C**, **D**. Three of the bit patterns are valid floating point numbers and one is not.

Figure 6



1 Complete **Table 3**. In the Correct letter (**A-D**) column shade the appropriate lozenge **A**, **B**, **C** or **D** to indicate which bit pattern in **Figure 6** is an example of the type of value described in the Value description column.

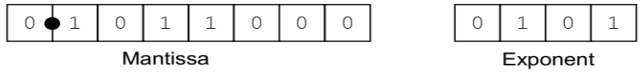
Do **not** use the same letter more than once.

[3 marks]

Table 3

Value description	Correct letter (A-D)
A positive normalised value	A ⬭ B ⬭ C ⬭ D ⬭
The most negative value that can be represented	A ⬭ B ⬭ C ⬭ D ⬭
A value that is not valid in the representation because it is not normalised	A ⬭ B ⬭ C ⬭ D ⬭

2 The following is a floating point representation of a number:



Calculate the decimal equivalent of the number. Show how you have arrived at your answer.

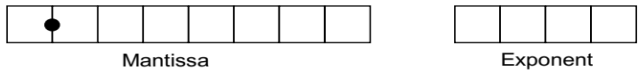
[2 marks]

Answer:

3 Write the normalised floating point representation of the negative decimal value -6.75 in the boxes below. Show how you have arrived at your answer.

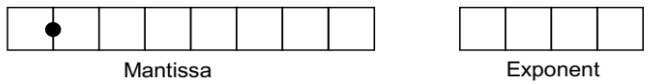
[3 marks]

Answer:



An alternative two's complement format representation is proposed. In the alternative representation 6 bits will be used to store the mantissa and 6 bits will be used to store the exponent.

Existing Representation (8-bit mantissa, 4-bit exponent):



Proposed Alternative Representation (6-bit mantissa, 6-bit exponent):

4 Explain the effects of using the proposed alternative representation instead of the existing representation.

[2 marks]

Q2:

1 Shade the lozenges next to **all** of the **true** statements about representing numbers using fixed and floating point representations.

[2 marks]

A A processor can usually carry out calculations on fixed point numbers more quickly than calculations on floating point numbers. ⬭

B Fixed point numbers represent data using a mantissa and an exponent. ⬭

C In a given number of bits, a fixed point system can represent positive numbers that are closer to zero than a floating point system can. ⬭

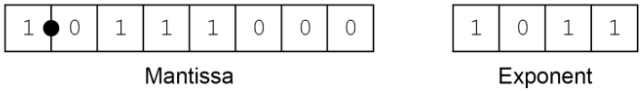
D In a given number of bits, a fixed point system can represent some numbers more precisely than a floating point system. ⬭

E In a given number of bits, a floating point system can represent a bigger range of numbers than a fixed point system. ⬭

Questions **04.2**, **04.3** and **04.4** use a **normalised** floating point representation with an **8-bit mantissa** and a **4-bit exponent**, both stored using **two's complement**.

2 **Figure 2** shows a floating point representation of a number:

Figure 2



Calculate the decimal equivalent of the number in **Figure 2**.

You should show your working.

[2 marks]

Answer _____

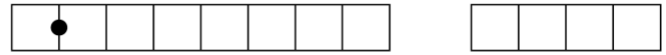
Questions **04.2**, **04.3** and **04.4** use a **normalised** floating point representation with an **8-bit mantissa** and a **4-bit exponent**, both stored using **two's complement**.

3 The decimal number 12.765625 (which can also be expressed as $12\frac{49}{64}$) cannot be represented exactly in this floating point system.

Write the closest possible normalised representation of this number in the boxes below.

You should show your working.

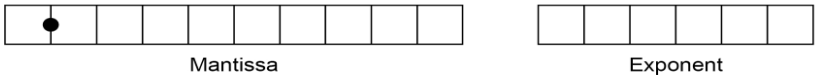
[3 marks]

Answer 

4 What is the smallest number of bits that would need to be **added to** the mantissa so that the decimal number 12.765625 could be represented exactly?

[1 mark]

5 A different system uses a **normalised** floating point representation with a **10-bit mantissa** and a **6-bit exponent**, both stored using **two's complement**.



In **decimal**, what is the most negative number that this system could represent?

You should show your working.

[2 marks]

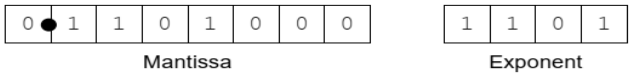
Most negative number _____

Q3

Question parts **06.1** and **06.2** use a **normalised** floating point representation with an **8-bit mantissa** and a **4-bit exponent**, both stored using **two's complement**.

1 **Figure 7** shows a floating point representation of a number:

Figure 7



Calculate the decimal equivalent of the number in **Figure 7**.

You should show your working.

[2 marks]

Answer _____

Question parts **06.1** and **06.2** use a **normalised** floating point representation with an **8-bit mantissa** and a **4-bit exponent**, both stored using **two's complement**.

2 Write the normalised floating point representation of the decimal value –23.25 in the boxes below.

You should show your working.

[3 marks]

Answer

3 On each row of **Table 1**, state the name of the **Type of error** that has occurred in the **Situation** that is described.

[2 marks]

Table 1

Situation	Type of error
A calculation is performed and the result of the calculation is so close to zero that the number that is stored is zero.	
A calculation is performed and the result of the calculation is too large to fit in the available number of bits.	
A decimal value is converted to floating point but it cannot be represented exactly in the available number of bits.	

Question parts **06.1** and **06.2** use a **normalised** floating point representation with an **8-bit mantissa** and a **4-bit exponent**, both stored using **two's complement**.

4 Explain how the floating point representation used in Question parts **06.1** and **06.2** could be modified to represent numbers more precisely, without changing the total number of bits used to represent a number.

[1 mark]

_____

_____

_____

Solutions

Q1

1	**All marks AO1 (understanding)** **1 mark** per correct response: **Value description** / **Correct letter (A-D)** A positive normalised value. — A The most negative value that can be represented. — C A value that is not valid in the representation because it is not normalised. — B If a letter is used more than once then mark as correct in the position where it is correct (if any).	3

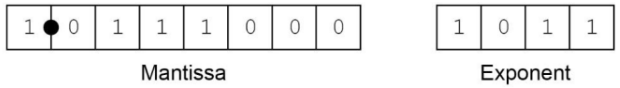
2	**All marks AO2 (apply)** 0 ● 1 0 1 1 0 0 0 — Mantissa 0 1 0 1 — Exponent **1 method mark** for either: • showing correct value of both mantissa and exponent in denary (Mantissa = 0.6875 // 11/16, Exponent = 5) • showing binary point shifted 5 places to right in binary number • indicating that final answer calculated using answer = mantissa x $2^{exponent}$ **1 mark** for correct answer Answer = 22 **If answer is correct and some working has been shown, award two marks, even if working would not have gained credit on its own.**	2

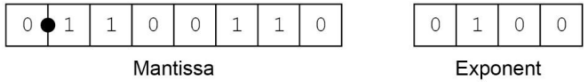
3	**All marks AO2 (apply)** **2 marks** for working: Correct representation of 6.75 in fixed point binary: 110.11; **A.** leading 0s. Correct representation of -6.75 in two's complement fixed point binary: 1001.01; **A.** leading 1s. Showing the correct value of the exponent in denary (3) or binary (11) // showing the binary point being shifted 3 places; **Max 2** **1 mark** for correct mantissa and exponent together: 1 ● 0 0 1 0 1 0 0 — Mantissa 0 0 1 1 — Exponent **If answer is correct and some working has been shown, award three marks, even if working would not have gained credit on its own.** **Working marks can be awarded for work seen in the final answer eg correct exponent.**	3

4	**All marks AO1 (understanding)** **1 mark:** Reduced precision; **1 mark:** Increased range; **A.** can represent larger/smaller numbers	2

Q2

1	**All marks AO1 (understanding)** **A** A processor can usually carry out calculations on fixed point numbers more quickly than calculations on floating point numbers. ⬬ **B** Fixed point numbers represent data using a mantissa and an exponent. ⬭ **C** In a given number of bits, a fixed point system can represent positive numbers that are closer to zero than a floating point system can. ⬭ **D** In a given number of bits, a fixed point system can represent some numbers more precisely than a floating point system. ⬬ **E** In a given number of bits, a floating point system can represent a bigger range of numbers than a fixed point system. ⬬ **2 marks**: All five rows shaded / not shaded correctly **OR** **1 mark**: Three or four rows shaded / not shaded correctly	**2**

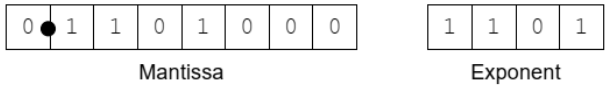
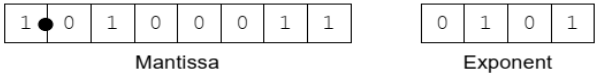
Pt	Marking guidance	Total marks
2	**All marks AO2 (apply)**  Award **2 marks** for correct answer: –0.017578125 // –9/512 **A.** rounded to at least 4 dp (eg –0.0176) If answer is incorrect then award **1 method mark** for either: • showing correct value of both mantissa and exponent in decimal (Mantissa = –0.5625 // –9/16 Exponent = –5) • showing binary point shifted 5 places to left in binary number • indicating that final answer has been calculated using answer = mantissa x $2^{exponent}$ and using either the correct mantissa, the correct exponent, or both in this calculation.	**2**

3	**All marks AO2 (apply)** Award **3 marks** for correct answer:  If answer is incorrect then award **up to 2 method marks**, one for each point from this list: • correct (unsigned) fixed point representation of 12.765625 in binary: 1100.110001; **A.** leading 0s and trailing 0s • correct 7-bit rounded fixed point representation of 12.765625 in binary: 1100.110; **A.** a single leading 0 • correct mantissa value: 0.1100110 • correct value of the exponent in decimal (4) or binary (100) // showing the binary point being shifted 4 places left **A.** if only shown in final answer exponent box	**3**

Pt	Marking guidance	Total marks
4	**Mark is AO2 (apply)** 3 (bits);	**1**

Pt	Marking guidance	Total marks
5	**All marks AO2 (apply)** Award **2 marks** for correct answer: –2147483648 **//** $–2^{31}$ // $–1 \times 2^{31}$ If answer is incorrect then award **1 method mark** for **one** of: • giving the most negative value of mantissa 1.000000000 and exponent 011111 • giving the most negative value as a binary integer: 1000 0000 0000 0000 0000 0000 0000 0000	**2**

Q3.

1	**All marks AO2 (apply)**  Award **2 marks** for correct answer: 0.1015625 // 13/128 **A.** Rounded to at least 4 dp (eg 0.1016) If answer is incorrect then award **1 method mark** for either: • showing correct value of both mantissa and exponent in decimal (Mantissa = 0.8125 // 13/16 Exponent = –3) • showing binary point shifted 3 places to left in binary number • indicating that final answer has been calculated using answer = mantissa × $2^{exponent}$ and used either the correct mantissa, the correct exponent, or both in this calculation.	2
2	**All marks AO2 (apply)** Award **3 marks** for correct answer:  If answer is incorrect then award **up to 2 method marks**, one for each point from this list: • correct (unsigned) fixed point representation of 23.25 in binary: 10111.01; **A.** leading 0s and trailing 0s • correct fixed point representation of –23.25 in binary: 101000.11; **A.** leading 1s and trailing 0s • showing the correct value of the exponent in decimal (5) or binary (101) in the working space or in binary in the final answer box // showing the binary point being shifted 5 places left; • showing the correct value of the mantissa in binary in the working space or final answer box.	3

Pt	Marking guidance	Total marks
3	**All marks AO1 (knowledge)** **1 mark** for two types of error correctly identified or **2 marks** for all three types of error correctly identified. **Situation** / **Type of error** A calculation is performed and the result of the calculation is so close to zero that the number that is stored is zero. — Underflow A calculation is performed and the result of the calculation is too large to fit in the available number of bits. — Overflow **R.** stack overflow A decimal value is converted to floating point but it cannot be represented exactly in the available number of bits. — Rounding **A.** truncation	2
4	**Mark is AO1 (understanding)** Move a/some bit(s) from the exponent to the mantissa; **A.** Increase number of bits in mantissa and reduce number of bits in exponent **A.** Examples in which mantissa has more than 8 bits and total number of bits in mantissa and exponent sums to 12 **NE.** Add more bits to the mantissa **NE.** Make the mantissa larger and the exponent smaller Use an implicit bit in the mantissa // do not store one of the bits on either side of the binary point as the value of the bit on one side of it can be inferred from // is the opposite of the value of the bit on the other side of it; **R.** use fixed point **Max 1**	1