Contents

[For the Exam: 1](#_Toc342402227)

[Revision of Binary 1](#_Toc342402228)

[Computing Mathematics Revision 1](#_Toc342402229)

[Binary Addition/Subtraction revision 2](#_Toc342402230)

[BCD 2](#_Toc342402231)

[Sign and Magnitude 2](#_Toc342402232)

[2’s Compliment 3](#_Toc342402233)

[Floating Point Representation 3](#_Toc342402234)

# For the Exam:

* Demonstrate an understanding of floating point representation and of a real binary number
* Terms should include a mantissa and exponent, Binary numbers may be in sign and magnitude or two’s compliment notation
* Normalise a real or binary number
  + Include the reason for normalisation
  + Be able to normalise positive or negative numbers
* Discuss the trade-off between accuracy and range when representing numbers
  + Relate to mantissa and exponent

# Revision of Binary

How do you convert from Denary to binary, describe the steps

Convert the following numbers into their proper number bases.

|  |  |  |  |
| --- | --- | --- | --- |
| **Hexadecimal** | **Binary** | **Octal** | **Denary** |
| A | 1010 |  |  |
| 2F | 101111 |  |  |
| 4C | 1001100 |  |  |
|  | 11011 |  | 27 |
|  | 10001100 |  | 140 |
|  |  |  | 99 |
| 3F |  | 077 |  |
| AB |  |  |  |
|  |  | 470 | 312 |

# Computing Mathematics Revision

What are the outcomes of adding the following numbers together in Binary

|  |  |
| --- | --- |
| Numbers | Outcome |
| 1+0 |  |
| 0+1 |  |
| 1+1 |  |
| 0+0 |  |

What are the steps to successfully add two numbers together in Binary?

## Binary Addition/Subtraction revision

Show the results of adding and subtracting the following numbers, Show your working

|  |  |  |
| --- | --- | --- |
| Numbers | Addition | Subtraction |
| 24  12 |  |  |
| 45  9 |  |  |
| 78  54 |  |  |
| 43  16 |  |  |

## BCD

|  |
| --- |
| What does BCD stand for? |
|  |
| How is it used? |
|  |

|  |  |
| --- | --- |
| Work out the BCD Representations for the following numbers: | |
| 1010 |  |
| 101111 |  |
| 1001100 |  |
| 11011 |  |
| 10001100 |  |
| 27 |  |
| 140 |  |
| 99 |  |

## Sign and Magnitude

|  |  |
| --- | --- |
| Sign and Magnitude – How is it used? | |
|  | |
| What are the disadvantages of sign and magnitude? | |
|  | |
| Work out the sign and magnitude for the following numbers | |
| -27 |  |
| 140 |  |
| -99 |  |
| 46 |  |

## 2’s Compliment

|  |  |
| --- | --- |
| What is 2’s Compliment? | |
|  | |
| How is it used? | |
|  | |
| Work out the 2’s Compliment for the following numbers | |
| -27 |  |
| 140 |  |
| -99 |  |
| 46 |  |

# Fixed Point Binary

So at the moment we’ve only dealt with whole numbers, now, this can become pretty easy after a while, so what about fixed point?

Fixed point in binary is still has the point as in decimal, but now we have 1’s and 0’s to deal with:

1010.1010

The stuff before the decimal point is easy, it’s 10, but what about the numbering system after the decimal point?

Note down below the numbering system after the decimal point

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
| 1 | 0 | 1 | 0 |

## Examples

Work out the following in fixed point representations:

|  |  |
| --- | --- |
| Number | Representation |
| 1101.1101 |  |
| 3.0625 |  |
| 9.6875 |  |
| 1110.0111 |  |

# Floating Point Representation

Before this point we have dealt with whole numbers, Positive and negative, yes, but floating point, no. This can be a problem as if we are ever doing a computer program which handles money, at some point you are going to have to write some code that deals with the dreaded 99p!

## Conversion of Floating Point Numbers

Floating point representation needs to use the 2’s compliment notation, meaning that the furthermost digit represents positive/negative numbers,

How is data represented in Floating Point

10010

0010

Exponent

Mantissa

|  |
| --- |
| What is the mantissa? |
|  |
| What is the exponent? |
|  |
| Explain the steps in order to convert the above number into a denary number |
|  |

## Exercises:

|  |
| --- |
| A number is represented as an 8 bit floating point number. 5 Bits are used as the mantissa, 3 bits are used as the exponent. Both the mantissa and the exponent are in twos compliment notation.  0.1100010 |
|  |
| 1.0010010 |
|  |

## Floating point range

Depending on the computer depends on the limitations of floating point. A 32 Bit computer can represent how many bytes in Floating point?

As with all computing it is a line of best fit. You can’t make every number in the known floating point universe, why?

What would happen if I had a 64 bit computer?

### Overflow and Underflow

|  |
| --- |
| What is Overflow? |
|  |
| What is Underflow? |
|  |
| What can we do about the overflow and underflow? |
|  |

## Normalisation of Binary Numbers

# Floating Point Binary – Normalisation

The precision of a floating point number depends on the number of digits stored in the mantissa but also on how these digits are used:

0.0035 x 108 = 350,000

0.3457 x 106 = 345,700 is more precise

In binary:

0 000110101 000110 becomes 0 110101000 000011

+ .000110101 exp=6 becomes + .110101000 exp=3

Normalisation is the process of moving the binary point so that the first digit after the point is a significant digit. This maximises precision in a given number of bits.

* To maximise the precision of a positive number you should have a mantissa with no leading zeros.
* To maximise the precision of a negative number you should have a mantissa with no leading ones.
* This means that the first two digits will always be different in a normalised floating point binary number.

**You try:**

1. Normalise the following **positive** numbers (10-bit mantissa and a 6-bit exponent) :
2. 0 000000110 000111
3. 0 000010111 000110
4. Normalise the following **negative** numbers (10-bit mantissa and a 6-bit exponent) :
5. 1 111110111 000000
6. 1 111111010 000011

# Floating Point Binary Practice questions

Typical exam question:

1. What is the decimal value for the following binary patterns if the first 8 bits is the mantissa and the last 4 bits are the exponent: 1000 1100 0100

How to do these questions!

|  |  |
| --- | --- |
| 1000 1100 0100  mantissa exponent | Separate out the exponent from the mantissa  *Originally this number is given in two’s complement* ***Normalised Form***  *(most accurate representation possible for a given size of mantissa)* |
| 1000 1100 **0100**  +4 | Work out the exponent and write it down (you may get a mark for this part!). NB: first bit of exponent (like mantissa) is the sign bit.  *In normalise form the binary point always goes after the first significant digit of the mantissa. This number represents 1.00011 x 24* |
| 1000 1100 0100  0111 0011 (complement)  1+ (add 1)  0111 0100 | If the mantissa is negative then work out the complement (swap 1’s and 0’s) and add 1 |
| 0**.**1110100  becomes  01110**.**100  +4 | Move the binary point to its new position.  Right for positive exponent, left for negative exponent.  *Normalised form always starts with 2 bits that are different*  *e.g. 01 or 10.* |
| 1110.1 = **-14.5** | Work out the number in decimal.  Remember add the minus sign back in at the end! |

Now you try!

|  |  |
| --- | --- |
| 1. 0101 1001 0110 | 4. 0100 0000 1111 |
| 2. 1000 1100 0100 | 5. 1011 0110 0101 |
| 3. 0110 1110 0100 | 6. 0110 1101 0101 |