Pearson BTEC Levels 5 Higher Nationals in Engineering (RQF)

Unit 39: Further Mathematics

Unit Workbook 1

in a series of 4 for this unit

Learning Outcome 1

Applications of Number Theory

Contents

INTRODUCTION

Use applications of number theory in practical engineering situations

Number theory:

Bases of a number (Denary, Binary, Octal, Duodecimal, Hexadecimal) and converting between bases.

Types of numbers (Natural, Integer, Rational, Real, Complex).

The modulus, argument and conjugate of complex numbers.

Polar and exponential forms of complex numbers.

The use of de Moivre's Theorem in engineering.

Complex number applications e.g. electric circuit analysis, information and energy control systems.

1 Number theory:

1.1 Number Types and Systems

The basic types of number we deal with regularly in engineering are:

- **Natural** – Everyday positive whole counting numbers (i.e. 0, 1, 2, 3, 4, 5, …)
- **Integer** – Similar to Naturals, but negatives are allowed (i.e. …-3, -2, -1, 0, 1, 2, 3, …)
- **Rational** – Can be expressed as a fraction of two integers (0 not allowed in denominator, i.e. $-9/5$)
- **Irrational** - Can NOT be expressed as a fraction of two integers (i.e. $\sqrt{2}$)
- **Real** – ANY quantity along a continuous line (i.e. -16.76449)
- **Complex** – Have real and imaginary components. Read more on these later

Common number systems encountered in engineering are:

- **Binary** – Base 2
- **Octal** – Base 8
- **Denary** – Base 10 (what you're most used to)
- **Duodecimal** (also known as **Dozenal**) – Base 12
- **Hexadecimal** – Base 16

To work efficiently in modern engineering, you need to be able to not only appreciate each of these number systems but also convert between them. You can always check your answers by using the Windows Calculator (click on View then Programmer). Using the calculator is fine, but Assignment 1 asks that you demonstrate knowledge of the mechanics of these conversions. Let's look at each system first and then work out how to go about the conversions in detail.

Binary System (Base 2)

The Binary system only uses two digits, 0 and 1 (hence the term **Bi**nary). Let's look at a template for the Binary system…

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
How many 128's	How many 64's	How many 32's	How many 16's	How many 8's	How many 4's	How many 2's	How many 1's

Octal System (Base 8)

The Octal system uses eight digits, 0 through to 7 (hence the term **Oct**al). Let's look at a template for the Octal system…

8^7	8^6	8^5	8^4	8^3	8^2	8^1	8^0
How many 2,097,152's	How many 262,144's	How many 32,768's	How many 4,096's	How many 512's	How many 64's	How many 8's	How many 1's

Denary System (Base 10)

The Denary system uses ten digits, 0 through to 9 (hence the term **Den**ary). Let's look at a template for the Denary system (which you got used to as a child) …

10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0
How many 10 millions	How many millions	How many 100,000's	How many 10,000's	How many 1,000's	How many 100's	How many 10's	How many 1's

Duodecimal System (Base 12)

The Duodecimal system uses twelve digits, 0 through to 9, then 'T' for ten and 'E' for eleven (hence the alternative term **Dozenal**, meaning dozen). Let's look at a template for the Duodecimal system …

12^7	12^6	12^5	12^4	12^3	12^2	12^1	12^0
How many 35,831,808's	How many 2,985,984's	How many 248,832's	How many 20,736's	How many 1,728's	How many 144's	How many 12's	How many 1's

Hexadecimal System (Base 16)

The Hexadecimal system uses sixteen digits, 0 through to 9 plus A through to F (hence the term **Hex**a**dec**imal). To employ 16 digits, we have used the letters A to F to represent the numbers 10 to through to 15 respectively. Let's look at a template for the Hexadecimal system…

16^7	16^6	16^5	16^4	16^3	16^2	16^1	16^0
How many etc.	How many etc.	How many etc.	How many 65,536's	How many 4,096's	How many 256's	How many 16's	How many 1's

Representing Numbers in Each System

If we wanted to represent the everyday Denary (base 10) number 12 in Binary format then we would have to work from left to right in the Binary table…

How many 128's in 12? Answer: 0 so put 0 in that cell

How many 64's in 12? Answer: 0 so put 0 in that cell

How many 32's in 12? Answer: 0 so put 0 in that cell

How many 16's in 12? Answer: 0 so put 0 in that cell

How many 8's in 12? Answer: 1 so put 1 in that cell and subtract the 8 from the 12, leaving 4

How many 4's in 4? Answer: 1 so put 1 in that cell and subtract the 4 from the 4, leaving 0

There is nothing left over here, so we've finished. We may now write $12_{10} = 00001100_2$.

Some examples of starting with a Denary number and representing it in the Binary, Octal and Hexadecimal formats now follow…

<u>Binary System (Base 2)</u>

Denary Examples	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
12	0	0	0	0	1	1	0	0
17	0	0	0	1	0	0	0	1
57	0	0	1	1	1	0	0	1
129	1	0	0	0	0	0	0	1

<u>Octal System (Base 8)</u>

Denary Examples	8^7	8^6	8^5	8^4	8^3	8^2	8^1	8^0
7	0	0	0	0	0	0	0	7
17	0	0	0	0	0	0	2	1
855	0	0	0	0	1	5	2	7
1,067,003	0	4	0	4	3	7	7	3

<u>Hexadecimal System (Base 16)</u>

Denary Examples	16^7	16^6	16^5	16^4	16^3	16^2	16^1	16^0
15	0	0	0	0	0	0	0	F
33	0	0	0	0	0	0	2	1
99,999	0	0	0	1	8	6	9	F
200,000,000	0	B	E	B	C	2	0	0

Conversions between bases:

Denary to Binary, Denary to Octal and Denary to Hexadecimal conversions are all performed in the manner described in the above tables.

Binary to Denary, Octal to Denary and Hexadecimal to Denary conversions are all performed in the reverse manner to those conversions already stated.

That leaves us with six types of conversion to deal with:

- Binary to Octal
- Octal to Binary
- Binary to Hexadecimal
- Hexadecimal to Binary
- Octal to Hexadecimal
- Hexadecimal to Octal

Binary to Octal Conversion

The quick way to do this is to group the Binary bits into groups of three and convert each group individually into Octal. A quick example…

101100010111 can be grouped as 101 100 010 111. The first group on the left is 101 which we know to be integer 5. Doing the same to the second group gives 4, then 2 for the third group and 7 for the last group. We can then say that $101100010111_2 = 5427_8$. Easy enough!

Octal to Binary Conversion

That's quite east too. Just turn each of the Octal digits in your number into their 3-digit binary equivalents, reversing the previous process. A quick example…

Octal 3627: convert the 3 to 011, convert the 6 to 110, convert the 2 to 010, convert the 7 to 111. This gives us $3627_8 = 011110010111_2$.

Binary to Hexadecimal Conversion

Just group your Binary bits into groups of four. A quick example…

101100010111 can be grouped as 1011 0001 0111. We now look at each group of four bits and convert that group to hex:

$$1011_2 = B_{16}, 0001_2 = 1_{16}, 0111_2 = 7_{16}$$

So, $101100010111_2 = B17_{16}$

Hexadecimal to Binary Conversion

Again, this is really easy. Just reverse the previous process.

Octal to Hexadecimal Conversion

Just convert your Octal Number into Binary bits – groups of 4 from the right-hand side. Then convert again to hex. Easy.