PROGRAMMABLE LOGIC CONTROLLERS

NQF LEVEL 3

OUTCOME 1
BASICS OF NUMBERING SYSTEMS AND NUMBER PROCESSING

This is set at the British Edexcel National level NQF 3

On completion of this tutorial you should be able to explain:

- Flags and Registers
- Number systems - Binary, octal, hexadecimal, binary-coded decimal (BCD); input/output devices

CONTENT

1. Introduction

Programmable logic controllers are digital devices. The internal structure is covered in outcome 2. All numbers presented for processing are in digital form and these may originate from software inside the PLC or from external devices connected to it. Information is stored in memory or on some other storage device as a series of digital numbers. This is just like an ordinary PC that you may be using that works internally on binary signals and communicates with your monitor, key board, mouse, modem and printer through binary numbers. Data is stored on your hard disc or CD or memory card as digital numbers.

At this level of study you need to understand the basic concepts of digital numbers only. Most scientific/engineering calculators are able to convert numbers from one system to another and you should make sure that you understand how to do this on your calculators. If you want to learn how to manipulate them arithmetically, you will find more advanced material on the web site www.freestudy.co.uk. Once this is mastered you can move on to seeing how they control a range of industrial processes.

2. Number Systems

Denary System

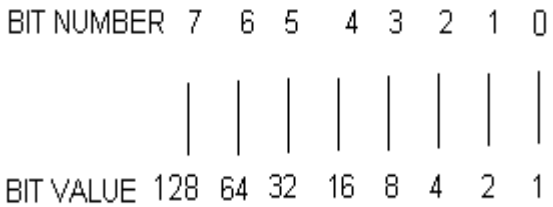
This is our normal system of numbering in tens.

Binary Numbers

A number may be represented in digital form by a simple pattern. The pattern may be generated for example with a row of lights switched on by an electric current. The pattern will also exist as a voltage level in the wires connected to the light bulbs. In digital electronics the pattern exists as a voltage relative to zero. Ideally in a typical computer, 5V is on or high and zero is off or low but to make a clear distinction, a voltage of over 3 is regarded high and below 2 is regarded as low. The patterns were originally developed in computers for 8 lines and then this became 16 then 32 and now it is 64.

Consider a pattern of 8 lines. We indicate on or high with a '1' and off or low with a '0'. Each line carries a bit of information as on or off and so the line is referred to as a bit.

In the denary system the digit that represents the highest value is on the left (e.g. the 3 in 32 or the 4 in 461). The digit representing the lowest value is on the right. These are called the most significant digit and least significant digit. In binary numbers we adopt the same idea with the bit on the left being the most significant bit (MSB) and the one on the right being the least significant bit (LSB).

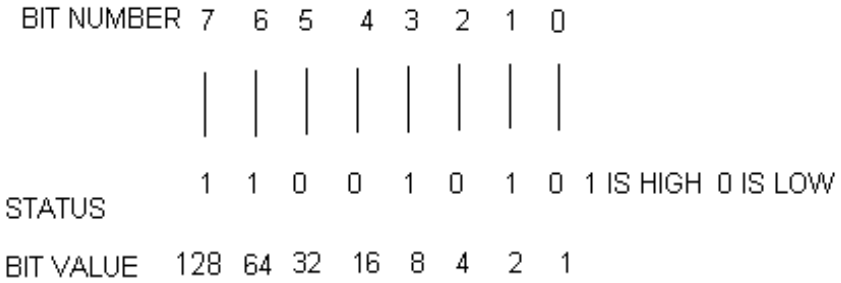


The total pattern is called a word and the one shown is an 8 bit word. The pattern may be stored in a register so it is also referred to as an 8 bit register. A ***register*** is a temporary store where the word may be manipulated.

Each bit has a value of zero when off (low) or the denary value shown when on (high). The denary value of the pattern is found by adding them all up. The maximum value for an 8 bit word is when all the bits are high and corresponds to 255.

WORKED EXAMPLE No. 1

What is the denary value of the digital pattern below?



SOLUTION

128 + 64 + 8 + 2 = 202

SELF ASSESSMENT EXERCISE No.1

Write down the decimal value represented by the following 8 bit patterns.

1 0 1 0 1 0 1 1 _____

1 0 0 1 0 0 1 0 _____

0 1 0 0 1 0 1 1 _____

Here is a small table of binary numbers with the equivalent denary values.

BINARY															
00	01	10	11	100	101	110	111	1000	1001	1010	1011	110	1101	1110	1111
DENARY															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

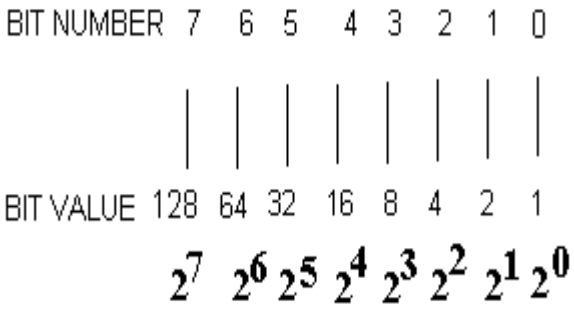
Conversion

A way of converting decimal into binary is to keep dividing the number by 2 as follows.
Convert decimal 12 into binary form.

12 ÷ 2 = 6 remainder 0
6 ÷ 2 = 3 remainder 0
3 ÷ 2 = 1 remainder 1
1 ÷ 2 = 0 remainder 1

The binary pattern is hence 1 1 0 0

Another way to express the value of each bit comes about by realising that each bit is 2 raised to the power of the bit number.



Octal and Hexadecimal

Octal is a numbering system based on cycles of 8. It is quite easy to convert a digital number into octal and for this reason it is used by programmers to address memory locations and port addresses (printer port, com port, CD driver address and so on). Early computers used words with 8 bits so octal was convenient to use.

Hexadecimal is a numbering system based on 16 and was introduced for the same reasons as octal when computers were developed with 16 bit registers. Hexadecimal Numbers are of far greater importance in modern computing. One Nibble (4 bits) is represented by 1 Hex digit. One Byte is represented by 2 Hex Digits.

In octal we have no need for the figures 8 and 9 as the cycle restarts after 7.
Octal numbers are indicated by the symbol @ e.g. @27

In hexadecimal, we need extra figures and the letters A, B, C, D, E and F are used.
Hexadecimal numbers are indicated by the symbol & (ampersand) e.g. &2DF

Here is the beginning of a conversion table:

Decimal	Binary	Octal	Hexadecimal
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
.	.	.	.
.	.	.	.
.	.	.	.

Binary Decimal Codes

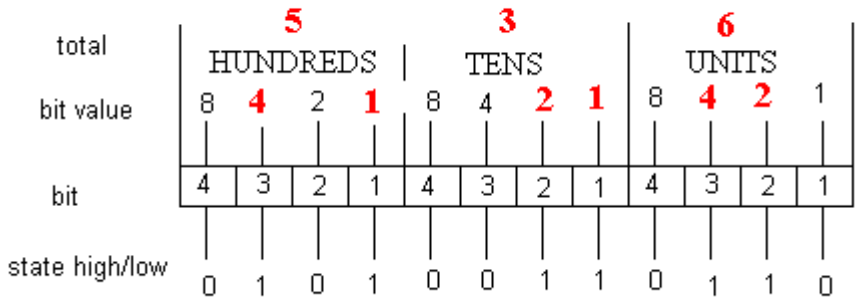
BCD is yet another way of presenting digital data as a number and it is used in industrial applications (e.g. to code or decode a digital signal from a position transducer).

The digital pattern in a register is converted into a 3 digit BCD as follows. The first 4 bits are turned into a normal (denary) number and this number is the units. The next 4 bits are the tens and the next 4 bits are the hundreds. The units cannot exceed a total of 9. The tens cannot exceed a total of 90 and the hundreds cannot exceed a total of 900. The maximum number which can be represented is 999.

WORKED EXAMPLE No. 2

Deduce the value of the BCD pattern.

3 digit BCD 536



SOLUTION

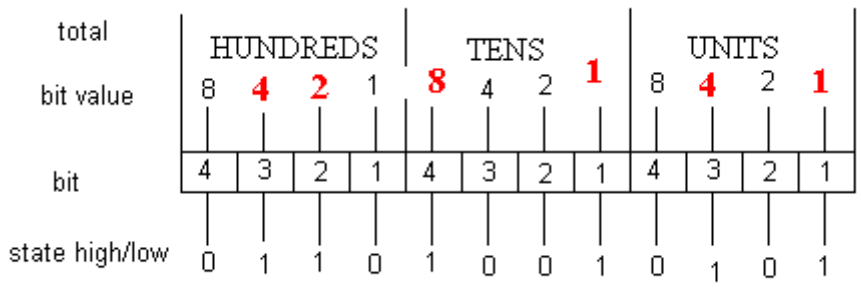
The first 4 bits decode to 4 + 2 = 6
The second 4 bits decode to 2 + 1 = 3
The third 4 bits decode to 4 + 1 = 5

The decimal value represented is hence 536

SELF ASSESSMENT EXERCISE No. 2

1. Use your calculator to convert the following denary numbers into binary, octal and hexadecimal: 45, 20, and 125
   (Answers for 45 are 101101, 55 and 2d you do the rest)

2. Deduce the value of this bcd pattern



(Answer 695)

Adding In Decimal

```
  235+
  926
1161
```

Starting with the LSB	5 + 6 = 11	write down 1 carry 1
Next	3 +2 + carry 1 =6	write down 6 carry 0
Next	2 + 9 + carry 0 = 11	write down 1 carry 1

Next start a new column and write down carry 1

Adding In Binary

The process is the same.

```
  01101101 +
  11011001
101000110
```

Starting the LSB,	1+1 = 10	write down 0 carry 1
Moving to bit 1	0+0 + carry = 1	write down 1 carry 0
Moving to bit 2	1+0 + carry 0 = 1	write down 1 carry 0
Moving to bit 3	1+1+ carry 0 = 10	write down 0 carry 1
Moving to bit 4	0+1+ carry 1 = 10	write down 0 carry 1
Moving to bit 5	1+0+ carry 1 = 10	write down 0 carry 1
Moving to bit 6	1+1+carry 1 = 11	write down 1 carry 1
Moving to bit 7	0+1+carry 1 = 11	Write down 1 carry 1

Moving to bit 8 which does not exist there is an implied 0 + 0 + carry 1 so write down 1. Clearly to store this number you would need a register with more than 8 bits.

Adding In Octal

This is similar but remember that it is based on 8.

```
332 +
167
521
```

Starting with the LSB	2 + 7 = 1 1	write down 1 carry 1
Next	3 + 6 + carry 1=12	write down 2 carry 1
Next	3+1+ carry 1 = 5	write down 5

Adding in HEX is more difficult as you need to remember the values of ABCDEF

It is best to do it on the calculator. Computers do it in binary and the answers are usually expressed in Decimal, Octal or Hex.

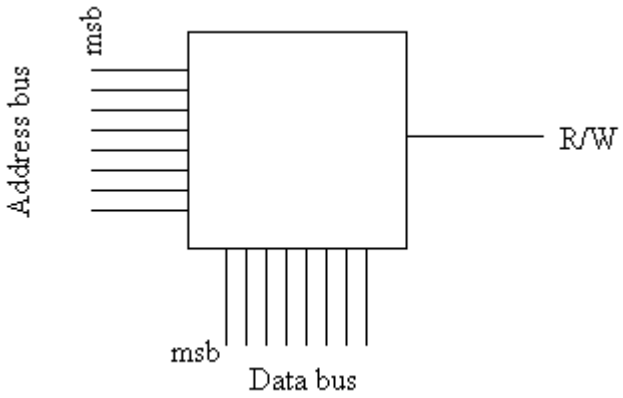
SELF ASSESSMENT EXERCISE No. 3

Use your calculator to do the following calculations.

Binary	Octel	Hexadecimal
11011 × 01101 =	@27 × @35=	&2D5 × &3A1=
11100 + 101100 =	@326 + @667=	&5CE + &9EF=
11000 – 01011 =	@642 – @341=	&FF5 – &3DE=

3. ***Applications and Manipulation***

Consider the basic operation of a computer memory ship. The chip has an address bus and a data bus. It also has a read or write command line.



Each line in the bus can be on (high) or off (low) so the system is based on binary patterns. Older equipment had 8 lines in each bus and so Octal became useful. Later this increased to 16 lines so hexadecimal became useful. It has since moved on to 32 and 64 lines.

Clearly to use larger numbers, you need more lines in the bus.

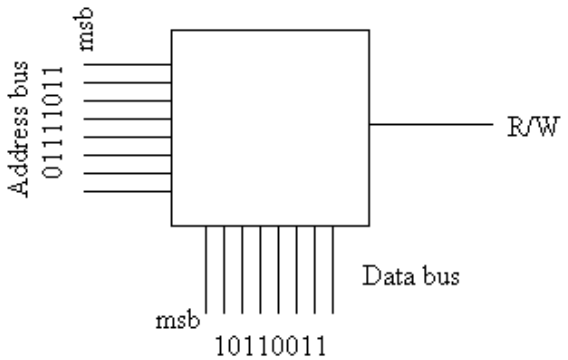
The diagram shows 8 lines in each bus. When they are all low the decimal value is 0. When they are all high the decimal value is 255 so there are 256 different values that can be represented.

When the R/W line is set to write, the binary pattern on the data bus is transferred into a store at the address represented by the binary pattern on the address line. This store is also called a register and the binary pattern is contained in this register.

When the R/W line is set to read, the pattern stored in the register at the address on the address bus is transferred to the data bus.

SELF ASSESSMENT EXERCISE No. 4

Write down the data value in decimal and the address in octal for the case shown.



Registers

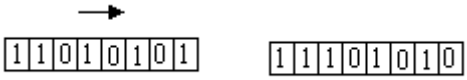
When a stored number is to be used in a calculation, it is stored in a special memory location called a **REGISTER**. Numbers stored in registers may be added or subtracted to a number stored in another register. This is done in a processing chip. The number from two memory locations is transferred into data registers. A command to the processor makes it add them and the result is placed in a third data register and perhaps transferred to a memory location.

Manipulation

The important difference between a register and any other memory location is that the bits may be manipulated under control of the programme. Hence the bits may be shifted right or left or rotated. Depending on the programme, the new bit may be a 1 or a 0.

Shifting

Consider the example below where the register is shifted right with a 1 being added.



What is the decimal value before the shift? _____

What is the decimal value after the shift? _____

Now consider the same problem with a left shift.

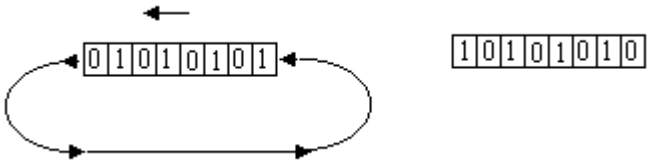
What is the decimal value before the shift? _____

What is the decimal value after the shift? _____

Each bit of the register may be used to actually switch something on or off so if a bit changes from 0 to 1 whatever is connected to the bit is turned on. There are many uses for shifting; one of them is to produce a counter.

Rotating

Rotating is similar to shifting but the bit that drops off the end reappears as the new bit.



Numbers stored in registers can be manipulated to produce actions such as adding or multiplying.

Adding and Subtraction

The contents of one register are added or subtracted to/from the contents of another and the result placed in a third register.

Multiplication and Division

The contents of one register may be multiplied or divided by the contents of another and the result placed in a third register.

Incrementing and Decrementing
The decimal value of the register is increased by 1 when incremented and decreased by 1 when decremented.

Comparison
The contents of two registers may be compared to see if they are the same.

Flags
These are single bits in a register that are switched on to indicate the status of something. They are not physical outputs. They can be addressed and used within a programme. PLCs often contain flags that are automatically switched on when a certain event happens.

Other arithmetical operations may be done such as square roots and differentiation depending on the PLC. The actual operation is much more complicated than discussed here and the result leads to carry over or borrowing of bits. There are special flags that are switched on to indicate these things.