CHAPTER 9:.. TABLE OF CONTENTS:

TABLE OF FIGURES:

9. Number Processing In The PLC.

9.1 Introduction to BINARY Numbers and their relationship with Denary (Decimal) Numbers.

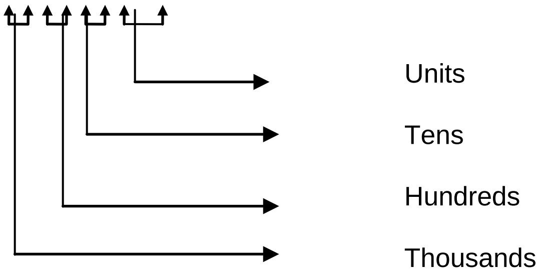
The denary or decimal system of numbers is the everyday system we work with. We have been taught to think in terms of these base 10 numbers. Digital computers employ a much similar system of numbers. They work with **BINARY** Numbers. It is very important that we develop an understanding of binary numbers so as we can begin to comprehend how computers and microprocessors work.

The binary system is the simplest of all number systems. Binary numbers contain only two different states “0” or “1” whereas decimal numbers contain 10, i.e. 0, 1, 2, 3, 4, 5, 6, 7 ,8, 9.The total number of different states used in a numbering system is known as the BASE or RADIX of the system. The base of the binary is 2 whilst the base of the decimal system is 10. Other numbering systems used in computers are Octal base 6 and Hexadecimal base 16.

9.1.1 Decimal System.

The decimal number 7,248 (seven thousand, two hundred and forty eight) stands for the following,

Decimal No. 7 2 4 8



From the above 7, 248 stands for 7 thousand + 2 hundreds + 4 tens + 8 units

Mathematically: $7*10^3 + 2*10^2 + 4*10^1 + 8*10^0 (10^0 = 1)$
$= 7x1000 + 2x100 + 4x10 + 8x1 = 7248$

Note: The BASE of the decimal system is 10.

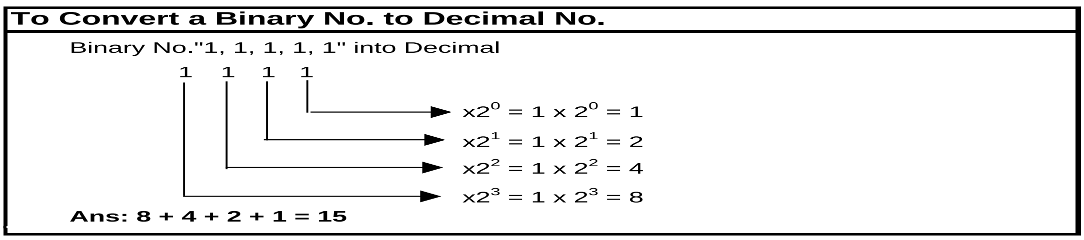
9.1.2 Binary System

Since PLC’s can only process the two signal states “0” and “1”. In the binary system we write numbers as a sequence of 1’s and 0’s. Zero in binary may be represented by a single “0” or a succession of “0” e.g 0, 00, 0000, all represent “0”.

1 represents powers of 2, i.e. 2^0, 2^1, 2^2, 2^3, 2^4. It is the position of a 1 digit in binary number that determines its value. For example the 4 digit binary number has the following meaning,

9.1.2.1 Convert Binary Number into Decimal Number

Convert the following binary number into the following decimal number,



CONVERT BINARY TO BINARY:

1 Bridge St,
Kilkelly,
Co. Mayo
Telephone: +353 (094) 9370934
Facsimile : +353 (094) 9370934
e-mail: training@focus-eng.com
Web: www.focus-eng.com

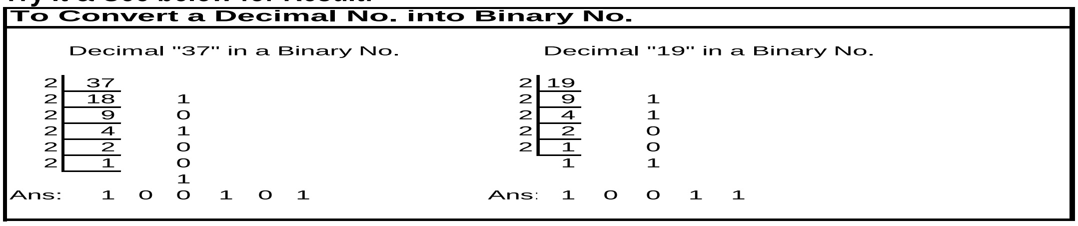
Note: Each Binary digit is referred to as a "**BIT**". The right-hand bit having the least weighing is known as the least significant bit (LSB) and the left-hand bit having the most weighing is known as the most significant bit (MSB). As we progress from right to left successive bits increase their weighing value in powers of 2.

For example convert the binary number 1 0 1 0 0 1 1 0 into a decimal number,

$1x2^7 + 0x2^6 + 1x2^5 + 0x2^4 + 0x2^3 + 1x2^2 + 1x2^1 + 0x2^0$
1x128 + 0x64 + 1x32 + 0x16 + 0x8 + 1x4 + 1x2 + 0
128 + o +32 +0 +0+ 4 +2+ 0 = 166

9.1.2.2 Convert a Decimal Number into a Binary Number

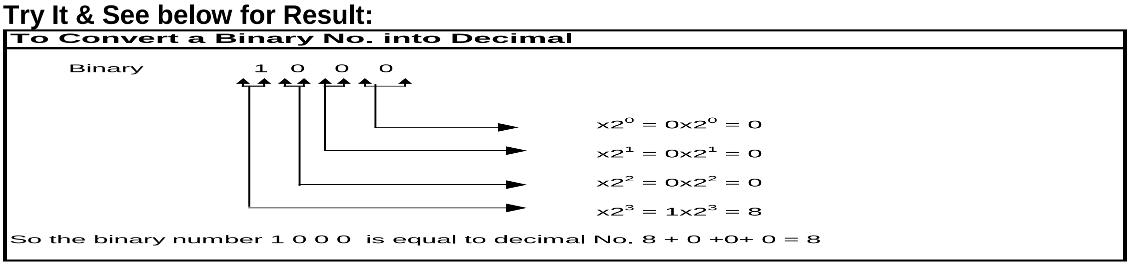
Convert the following decimal numbers 37 & 19 into the following binary number,
Try It & See below for Result:



CONVERT DECIMAL TO BINARY

9.1.3 Exercise Convert the following Binary Number in to Decimal Number.

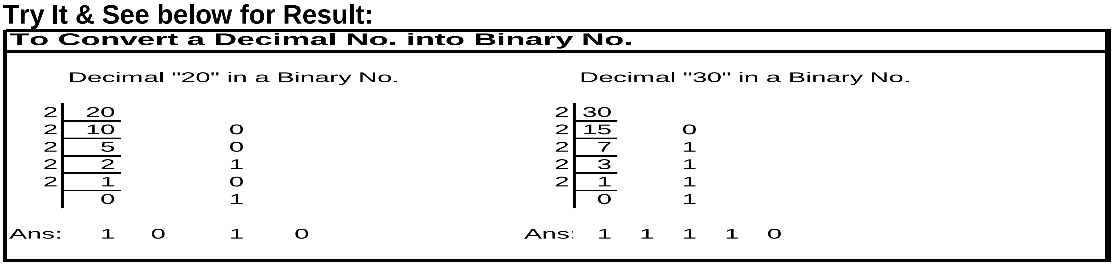
Convert the following Binary Number 1 0 0 0 into Decimal Number,
Try It & See below for Result:



EXERCISE CONVERT BINARY TO DECIMAL:

9.1.4 Exercise Convert the following Decimal Number in to Binary Number.

Convert the following Decimal Number's 20 & 30 into Binary Format,
Try It & See below for Result:



EXERCISE CONVERT DECIMAL TO BINARY:

9.1.5 Hexadecimal System

The Hexadecimal numbers are base 16 numbers employing 16 distinct digits to represent the decimal numbers 0, 1, 2, 3,……..12, 13, 14, 15. The least significant digit in front of the hexadecimal point is 16^0 = 1 (i.e. units), the next is 16^1, the next 16^2 and so on

There is however a problem of notation. Using the digits available in the decimal system what does one do to write down the double figures in the range 10 ….15. This is overcome by assigning the first six letters of the alphabet as follow,

Decimal Value	Hexadecimal Value
10	A
11	B
12	C
13	D
14	E
15	F

9.1.5.1 Convert hexadecimal to decimal

To convert a hexadecimal number to decimal one must complete the following as follows,

Convert Hexadecimal No. 20B7 (Hex) into Decimal No.

Hexadecimal "20B7" in a Decimal No.

	2	0	B	7
Ans:	$+2x16^3$	$+0x16^2$	$+11x16^1$	$+ 7x16^0$
Ans:	2x4096	+0x256	+11x16	+7x1
Ans:	8192	+0	+176	+7
Ans:	= 8375			

CONVERT HEXADECIMAL TO DECIMAL:

9.1.5.2 Convert decimal to hexadecimal

To convert a decimal Number into hexadecimal number the procedure is very similar to that of binary conversion etc. except that we divide the decimal number successively by 16

Convert the following Decimal Number's 268 into Hexadecimal Format,

To Convert a Decimal No. into Hexadecimal No.

Decimal "268" in a Hexadecimal No.

16	268	
16	16	12 =("C")
16	1	0
16	0	1
Ans:	1 0 C	

Decimal "30" in a Binary No.

16	36	
16	2	4
	0	2
Ans:	2 4	

CONVERT DECIMAL TO HEXADECIMAL:

9.1.5.3 Convert binary to hexadecimal

Binary to hex conversions are very similar to binary to decimal but to form the hexadecimal number equivalent we group the binary bits in groups of 4 and then replace each group by its hex equivalent

To convert the following binary number 1011100011110011 into hexadecimal complete the following,

Convert a Binary No. into Hexadecimal No.

Binary Number:	"1011100011110011" to Hexadecimal			
Groups of "4"	"1011, 1000, 1111, 0011"			
Decimal Equivalent:	11	8	15	3
Hexadecimal Equivalent:	B	8	F	3
Ans:	B	8	F	3

CONVERT BINARY TO HEXADECIMAL:

9.1.6 Exercise Convert the following Hexadecimal No. C4BA to Decimal.

Convert the following Hexadecimal Number C4BA into Decimal Format,

Try It & See below for Result:

Convert Hexadecimal No. C4BA (Hex) into Decimal No.

Hexadecimal "C4BA" in a Decimal No.

	C	4	B	A
Ans:	$+Cx16^3$	$+4x16^2$	$+Bx16^1$	$+ Ax16^0$
Ans:	$+12x16^3$	$+4x16^2$	$+11x16^1$	$+ 10x16^0$
Ans:	12x4096	+4x256	+11x16	+10x1
Ans:	49152	1024	+176	+10
Ans:	= 50,362			

EXERCISE CONVERT HEXADECIMAL TO DECIMAL:

9.1.7 Exercise Convert a Decimal Number to Hexadecimal Number.

Convert the following Decimal Number 46 and 100 into Hexadecimal Format,
Try It & See below for Result:

To Convert a Decimal No. into Hexadecimal No.					
Decimal "46" in to Hexadecimal No.			Decimal "100" in to Hexadecimal No.		
16	46		16	100	
16	2	E (=14)	16	6	4
	2	2	16	6	6
Ans: 2E			Ans:	64	

EXERCISE CONVERT DECIMAL TO HEXADECIMAL:

9.2 Decimal to Hexadecimal to Binary Reference Table,:

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	0001, 0000
17	11	00001, 0001

REFRENCE DECIMAL TO HEXADECIMAL TO BINARY

9.3 Number Representation in PLC (Bits, Bytes and Words):

Information about the signal states in the PLC is exchanged in **BITs**, **BYTEs**, **WORDs** and **DOUBLE WORDs**, 8 bits being grouped together to make one byte. In this case, we don't want these bits to represent signal states, but numbers. Each bit is to stand for one place value in a number, as in the binary system.

Note: If using Word Address the address of its most significant byte (high byte) should preferably be an even number.

Note: The largest number that can be represented using 8 bits of a Byte is 255. The largest number that can be represented using 16 bits of a Word is 65535.

To represent larger numbers, more bit positions are required. The S7-315 CPU can handle numbers that can be represented by a maximum of 16 bit positions.

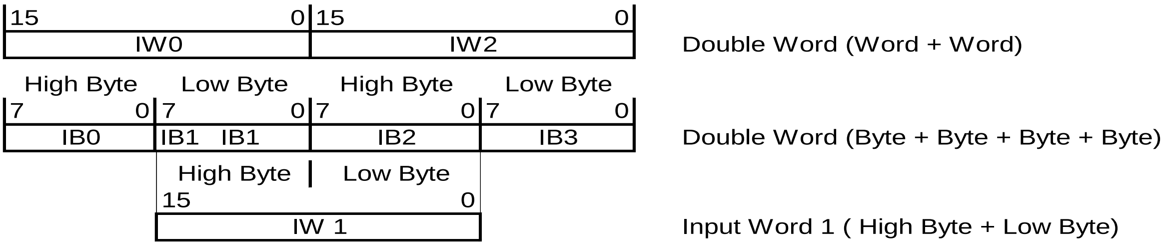
1 Bridge St,
Kilkelly,
Co. Mayo

Telephone: +353 (094) 9370934
Facsimile : +353 (094) 9370934
e-mail: training@focus-eng.com
Web: www.focus-eng.com

Example:
Input word IW 0 is made up of input bytes "**IB 0**" and "**IB 1**". IB 0 is the high byte and IB 1 the low byte. As far as the values of the bit positions are concerned, this means that :

Bit position I 0.7 has the value 2^{15} and bit position I 1.7 has the value 2^{7}.

Bit, Byte, Word & Double Word Addressing:



CONVERT DECIMAL TO HEXADECIMAL:

Note: Bit positions I 0.7 has the value 2^{15} and bit position I 1.7 has the value 2^{7}.

9.4 Edit the Force Variable

Using the Force Variable function the signal states can not only be displayed on at the PG terminal but also modified. This means that the function can be used to access and modify data (e.g. Digital Outputs) in the system data area of the CPU.

For example, if the status of an output which is connected to a motor is forced to '1' in the PIQ, the motor will start up (provided it is not reset again by the user program in the next cycle).

Note: When using this function, you must know exactly which parameters will be changed, so that you do not allow a dangerous situation to arise in the plant.

9.5 Processing and Storing Numbers in the PLC

9.5.1 Load and Transfer Operations:

Most of the operations you have met so far have consisted of checks and assignments, i.e. scanning and setting individual bits. For working with whole bytes or words the "**LOAD"** and "**TRANSFER"** operations are used.

Note: IN STL "LOAD" and "TRANSFER" operations are UNCONDITIONAL operations; i.e. the processor performs them in every cycle regardless of the RLO.

Load instruction
"**L IB**" or "**L IW**" causes the processor to fetch an input byte or input word from system memory (PII = process input image) and deposit it in accumulator 1. In this case, the PII is the source memory. The Load instruction can also be used for fetching output bytes and output words, flag bytes and flag words from system memory, as well as for loading constants.

Transfer instruction
"**T QB**" or "**T QW**" causes the processor to store the contents of accumulator 1 in system memory (PIQ = process output image). It can also be used to output to flag bytes or output to flag words. The PIQ in system memory is the destination memory.

9.5.2 The Accumulator:

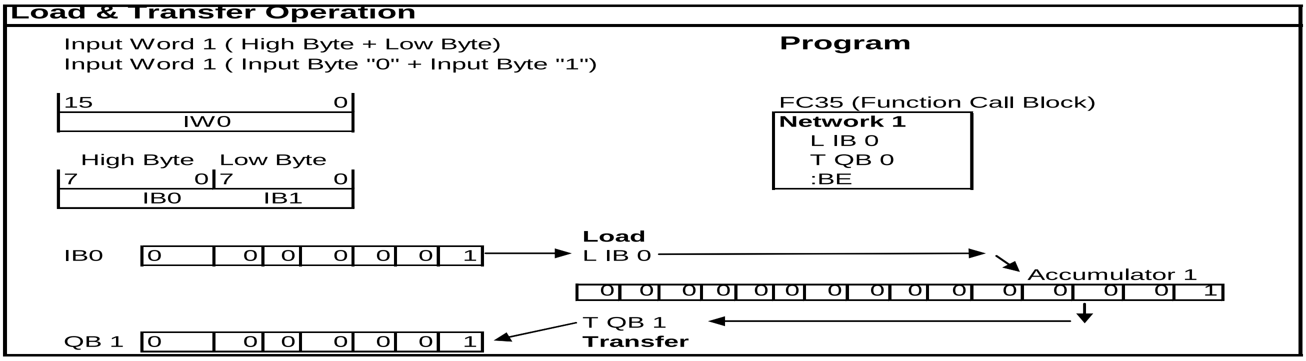
The processor normally has 2 accumulators ("**Accu 1**" and "**Accu 2**"). These are **REGISTERS** (Auxiliary Memories) in the processor which are used for copying data from one address to another, for comparisons and mathematical operations (**WORK MEMORY**). The direction of information flow depends on the instruction:

Load from source memory to accumulator 1. **Transfer** from accumulator 1 to destination memory.

In a **LOAD** operation, the contents of the relevant source memory are copied and written, into accumulator 1. **The previous contents of the accumulator 1 are shifted to accumulator 2.**

In a **TRANSFER** operation, the contents of accumulator 1 are copied and written to the relevant destination memory. The previous contents of the destination memory are overwritten. Since the contents of accumulator 1 are only copied, they are still there to be used in other transfer operations.

Note: The S7-300 has Two Accumulators with32 bits each and the S7-400 has Four Accumulators with 32 bits each.



LOAD AND TRANSFER OPERATIONS

9.6 Arithmetic Operations:

The operating system has an integral arithmetic block, which enables the processor to perform arithmetic operations:

Subtraction using the "**-I**" statement,

Addition using the "**+I**" statement.

Note: This arithmetic block works with 2 numbers, so it needs 2 accumulators: accumulators 1 and 2.

9.6.1 The Function of the Accumulators in Arithmetic Operations:

The user only has direct access to accumulator 1 when using the L (Load) and T (Transfer) instructions. Access to accumulator 2 is organized internally by the operating system.

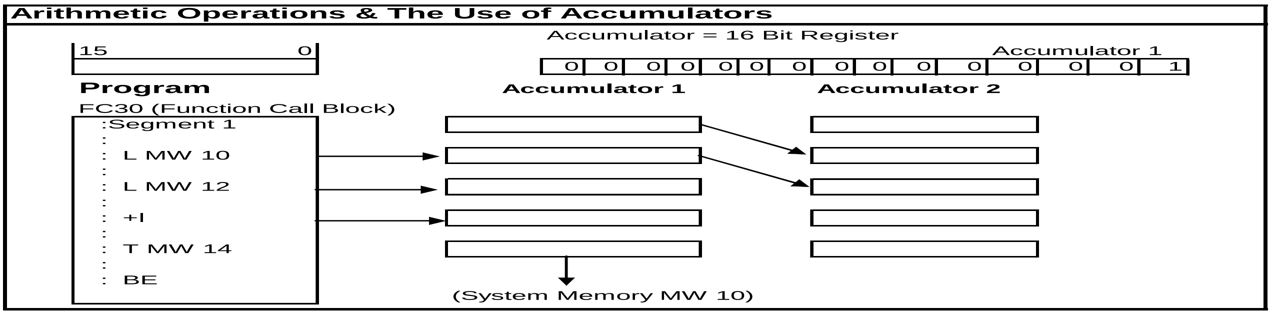
```
: L MW   10
: L MW   12
: +I
: T MW   14
```

The statement "**L MW10**" causes a copy of the flag word to be loaded, into accumulator 1. At the same time, the original contents of accumulator are shifted into accumulator 2. The statement "**L MW12**" causes the contents of accumulator 1 to be overwritten with a copy of **MW12**. The "**+I**" causes the contents of accumulator 1 to be added with the contents of accumulator 2.

Note: I means that the processor interprets and displays the values loaded (MW10 and MW12) as Integers (fixed-point numbers i.e. 12, 200, no decimal places).

The result of the addition is written to accumulator 1. The contents of accumulator 2 remains unchanged in this case (in the S7-315-2).

The statement "**T MW 14**" causes a copy of the contents of accumulator 1 (the result) to be stored in MW 14.



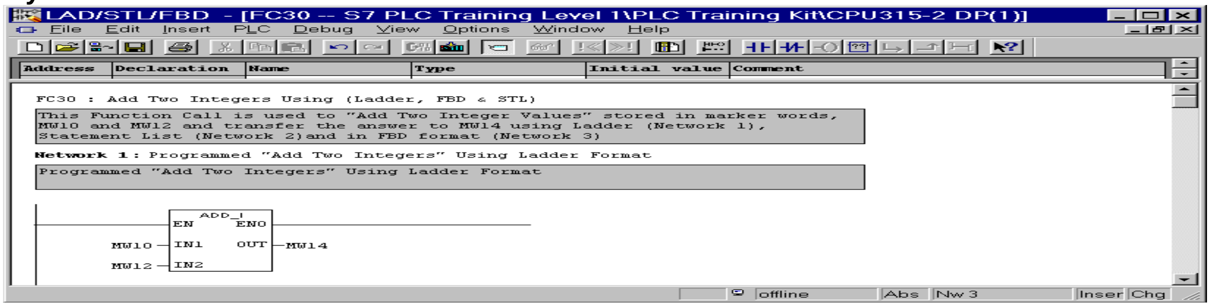
LOAD AND TRANSFER OPERATIONS

9.6.2 Exercise Create a FC30 in LAD Net, 1 “Add” MW10 + MW12 = MW14.

Write a program using Ladder Logic to “Add” MW10 and MW12 and transfer the answer to MW14,

Note: Force a value of 10 into MW 10, Force a value of 20 into MW 12

Try It & See below for Result:

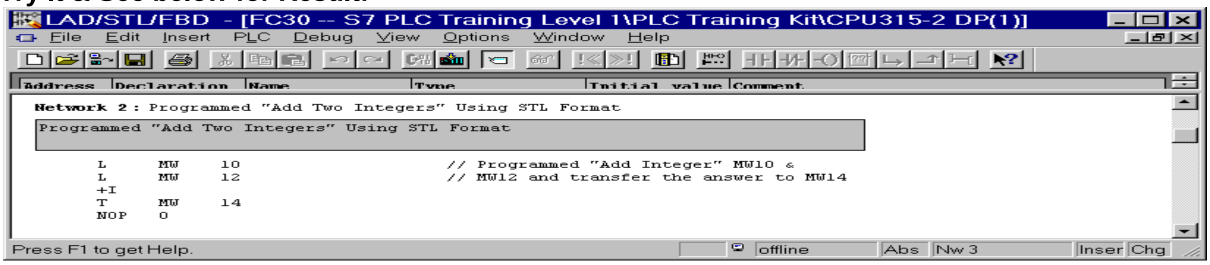


ADD TWO INTEGERS STORED IN MW10 & MW 12 USING LADDER LOGICI

9.6.3 Exercise Create FC30 in STL Net, 2 to “Add” MW10 + MW12 = MW14.

Write a program using STL Format to “Add” MW10 and MW12 and transfer the answer to MW14,

Try It & See below for Result:



ADD TWO INTEGERS STORED IN MW10 & MW 12 USING STL FORMAT

9.6.4 Exercise Create FC 30 in FBD Net,3 to “Add” MW10 + MW12 = MW14.

Write a program using FBD Format to “Add” MW10 and MW12 and transfer the answer to MW14,

Try It & See below for Result:

ADD TWO INTEGERS STORED IN MW10 & MW 12 USING FBD FORMAT

1 Bridge St,
Kilkelly,
Co. Mayo

Telephone: +353 (094) 9370934
Facsimile : +353 (094) 9370934
e-mail: training@focus-eng.com
Web: www.focus-eng.com

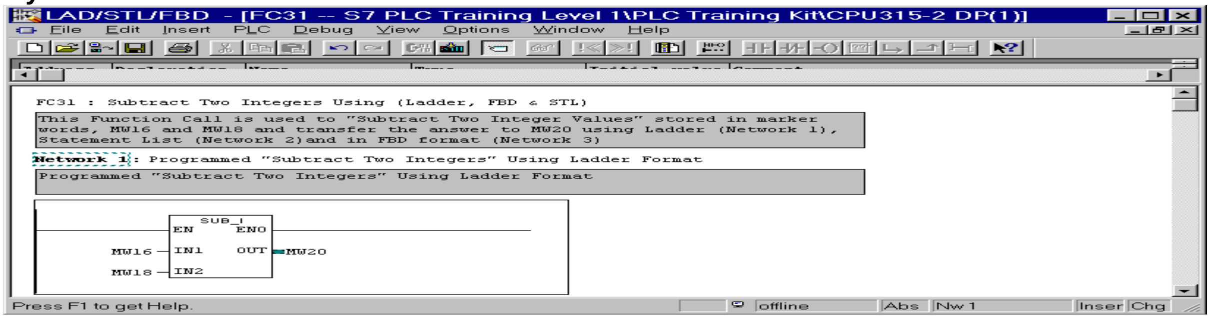
FOCUS ENGINEERING LTD REGISTRATION COMPANY NO. 283456W & VAT NO. IE8283456W

9.6.5 Exercise Create FC31 in LAD Net,1 to "Subtract" MW16 - MW18 = MW20.

Write a program using to "Subtract" MW16 from MW18 and transfer the answer to MW20,

Note: Force a value of 50 into MW 16 , Force a value of 100 into MW 18

Try It & See below for Result:

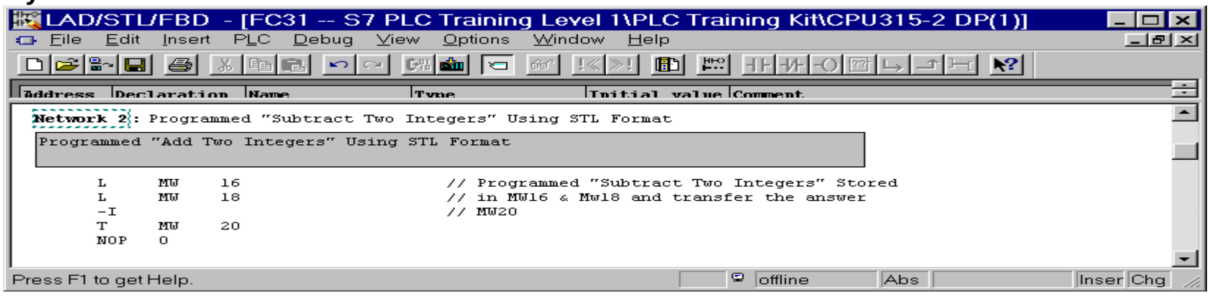


SUBTRACT TWO INTEGER VALUES STORED IN MW16 & MW 18 USING LADDER LOGICI

9.6.6 Exercise Create FC31 in STL Net,2 to "Subtract" MW16 - MW18 = MW20.

Write a program using STL to "Subtract" MW16 from MW18 and transfer the answer to MW20,

Try It & See below for Result:

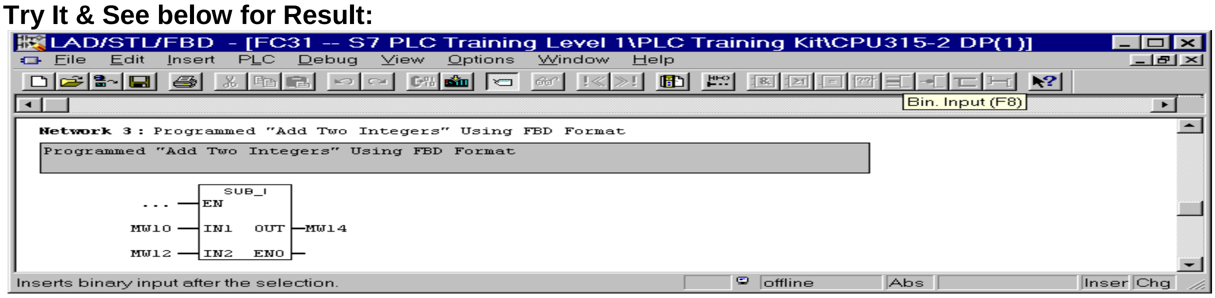


SUBTRACT TWO INTEGER VALUES STORED IN MW16 & MW 18 USING STL FORMAT

9.6.7 Exercise Create FC31 in FBD Net,3 to "Subtract" MW16 - MW18 = MW20.

Write a program using FBD to "Subtract" MW16 from MW18 and transfer the answer to MW20,

Try It & See below for Result:



SUBTRACT TWO INTEGER VALUES STORED IN MW16 & MW 18 USING FBD FORMAT

1 Bridge St,
Kilkelly,
Co. Mayo

Telephone: **+353 (094) 9370934**
Facsimile : **+353 (094) 9370934**
e-mail: **training@focus-eng.com**
Web: **www.focus-eng.com**

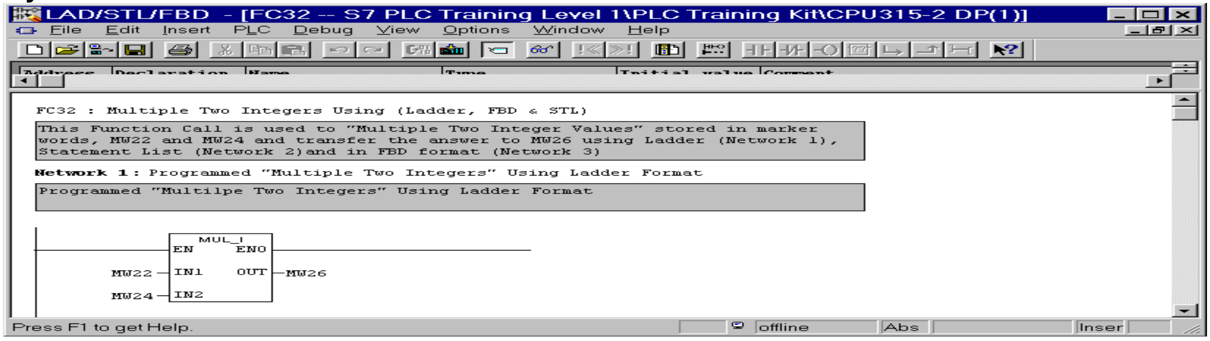
FOCUS ENGINEERING LTD REGISTRATION COMPANY NO. 283456W & VAT NO. IE8283456W

9.6.8 Exercise Create FC32 in LAD Net,1 to "Multiple" MW22 * MW24 = MW26.

Write a program using LAD to "Multiple" MW22 from MW24 and transfer the answer to MW26,

Note: Force a value of 10 into MW 22, Force a value of 5 into MW 24

Try It & See below for Result:

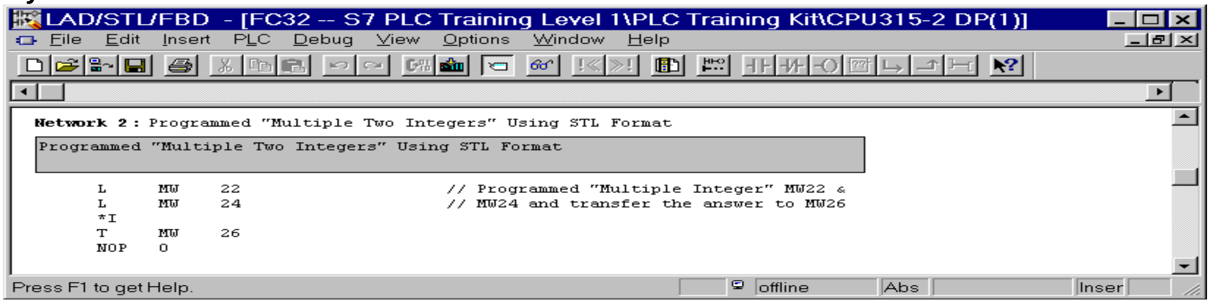


MULTIPLE TWO INTEGER VALUES STORED IN MW22 & MW 24 USING LADDER LOGICI

9.6.9 Exercise Create FC32 in STL Net,2 to "Multiple" MW22 * MW24 = MW26.

Write a program using STL "Multiple" MW22 to MW24 and transfer the answer to MW26,

Try It & See below for Result:

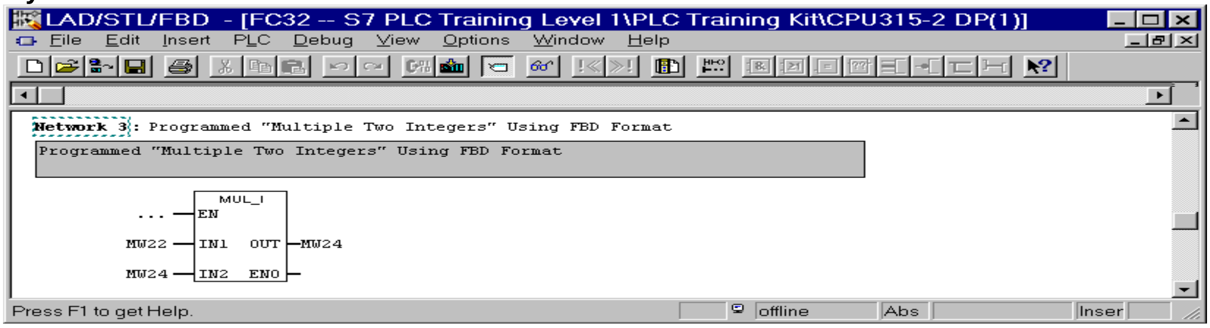


MULTIPLE TWO INTEGER VALUES STORED IN MW22 & MW 24 USING STL FORMAT

9.6.10 Exercise Create FC32 in STL Net,3 to "Multiple" MW22 * MW24 = MW26.

Write a program using FBD to "Multiple" MW22 to MW24 and transfer the answer to MW26,

Try It & See below for Result:



MULTIPLE TWO INTEGER VALUES STORED IN MW22 & MW24 USING FBD FORMAT

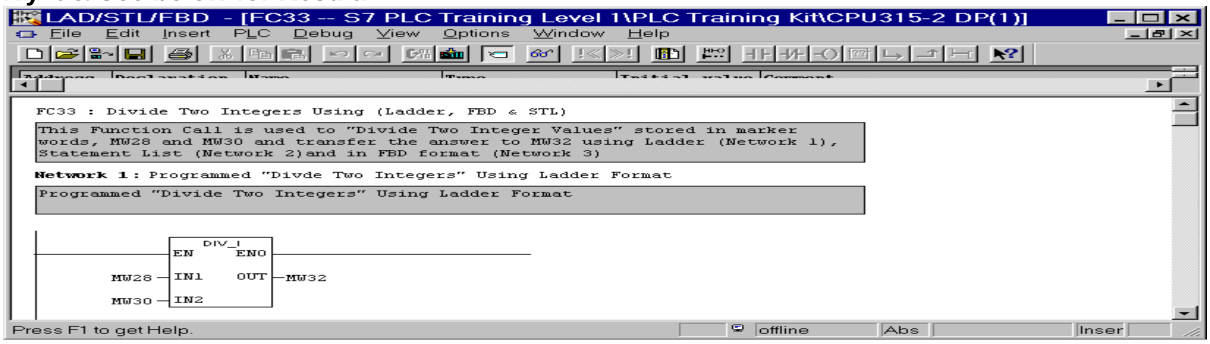
1 Bridge St,
Kilkelly,
Co. Mayo
Telephone: +353 (094) 9370934
Facsimile : +353 (094) 9370934
e-mail: training@focus-eng.com
Web: www.focus-eng.com
FOCUS ENGINEERING LTD REGISTRATION COMPANY NO. 283456W & VAT NO. IE8283456W

9.6.11 Exercise Create FC33 in LAD Net,1 to "Divide" MW28 / MW30 = MW32.

Write a program using LAD to "Divide" MW22 from MW24 and transfer the answer to MW26,

Note: Force a value of 100 into MW 28 , Force a value of 50 into MW 30

Try It & See below for Result:

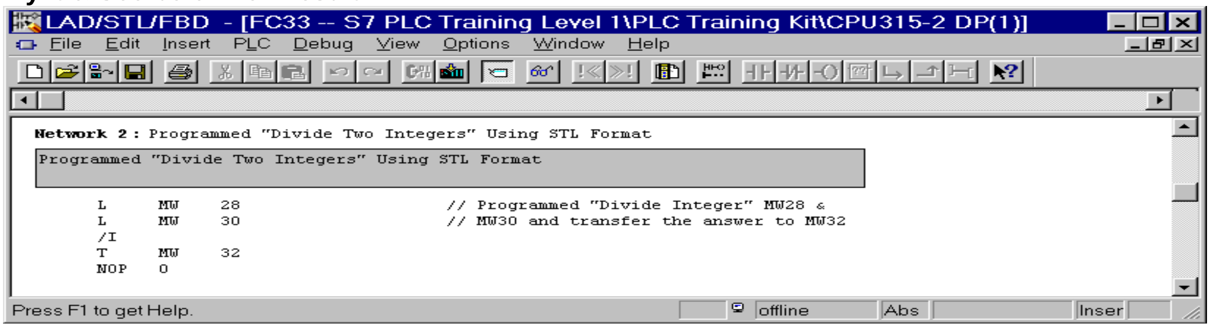


DIVIDE INTEGER VALUES STORED IN MW28 & MW 30 USING LADDER LOGICI

9.6.12 Exercise Create FC33 in STL Net,2 to "Divide" MW28 / MW30 = MW32.

Write a program using STL to "Divide" MW28 to MW30 and transfer the answer to MW32,

Try It & See below for Result:

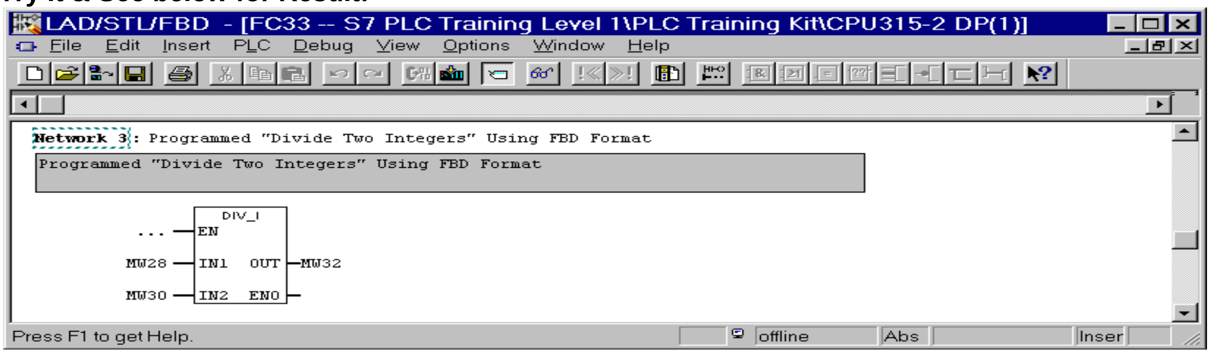


DIVIDE TWO INTEGER VALUES STORED IN MW28 & MW 30 USING STL FORMAT

9.6.13 Exercise Create FC33 in FBD Net,3 to "Divide" MW28 / MW30 = MW32.

Write a program using STL to "Divide" MW28 to MW30 and transfer the answer to MW32,

Try It & See below for Result:



DIVIDE TWO INTEGER VALUES STORED IN MW28 & MW30 USING FBD FORMAT

9.7 Exercise

Why use the Binary System ?, ____

____

Binary Numbers contain two different digit states what are they ?, _________________________

____

Convert the Binary No. 1,0,0,0,0,0,0 and 1,0,0,0,0,0,0,0 into decimal form ?. ____________________

____

Convert the decimal No's. 35 and 26 into Binary ?, ___________________________________

____

____

____

____

What is the difference between a Digital Signal and Binary Signal ?, _______________________

____

____

What is the difference between a Digital Signal and Analogue Signal ?, ________________________

____

____

What is the difference between a Analogue Signal and Binary Signal ?, ________________________

____

____

1 Bridge St,
Kilkelly,
Co. Mayo

Telephone: **+353 (094) 9370934**
Facsimile : +353 (094) 9370934
e-mail: training@focus-eng.com
Web: www.focus-eng.com

Check what you've learnt this Chapter

I Know:

⇒ What the Binary numbering system is.

⇒ What the Hexadecimal numbering system is.

⇒ How to convert from Decimal to Binary and Binary to Decimal.

⇒ How to convert from Decimal to Hexadecimal and Hexadecimal to Decimal.

⇒ How to convert from Hexadecimal to Binary and Binary to Hexadecimal.

⇒ What the formats KM, KF and KH is.

⇒ What the accumulators are used for within the CPU.

⇒ When accumulator 1 is modified when using the load instruction.

⇒ Which accumulator is used for storing the results of arithmetic operations.

⇒ How to "Add", "Subtract", "Multiple" and "Divide" Integers stored in Marker Words.

I Can:

⇒ Understand the structure of the Binary, Hexadecimal number system.

⇒ The difference between Decimal, Binary and Hexadecimal numbering system.

⇒ Convert from Decimal to Binary and Binary to Decimal.

⇒ Convert from Decimal to Hexadecimal and Hexadecimal to Decimal.

⇒ Convert from Hexadecimal to Binary and Binary to Hexadecimal.

⇒ Use the Arithmetic Function to Add, Subtract, Divide and Multiple.

⇒ Use the Force Variable Function in the Siemens PLCs.

⇒ How to "Add", "Subtract", "Multiple" and "Divide" Integers stored in Marker Words.