Chapter 15 Siemens FC and FB Programming

Revisit the Binary Addition/Binary Subtraction lab from chapter 8 to subtract one 16 bit word from another and put the 16 bit result in a third word using a function and using the Siemens TIA software.
Remember:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 carry 1
1 + 0 + carry = 0 carry 1
1 + 1 + carry = 1 carry 1

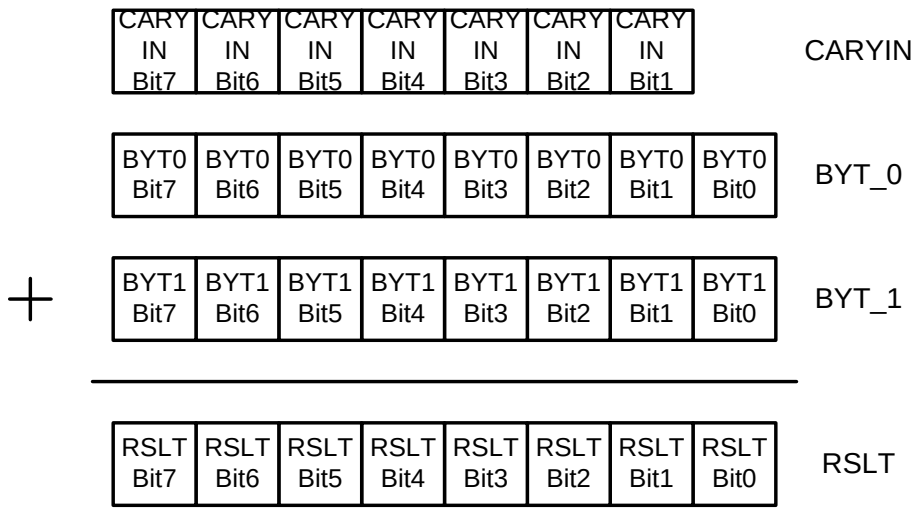
These are the rules for binary addition.

To see binary addition at work:

Carry				1	1	1	1				
Number 1	0	1	0	0	1	1	0	1	1	0	0
+ Number 2	0	1	0	1	1	0	1	1	0	1	0
Results	1	0	1	0	1	0	0	0	1	1	0

Binary addition may take place in ladder logic. Instructions are provided to carry out this function (ADD), but it is worthwhile to examine the process of binary addition using ladder logic

Since Bit 0 does not have a *carry_in*, half-adder logic may be employed but only for this bit. It can be seen that half-adder logic is simpler than full-add logic by comparing Chapter 8's Fig. 8-35 (Half-Adder) to Fig. 8-36 (Full Adder).

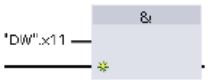


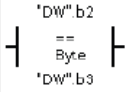
Accessing Bits in Words (Siemens)

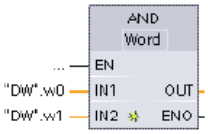
Examples

In the PLC tag table, "DW" is a declared tag of type DWORD. The examples show bit, byte, and word slice access:

	LAD	FBD	SCL

Bit access			`IF "DW".x11 THEN`
			`...`
			`END_IF;`

Byte access			`IF "DW".b2 = "DW".b3`
			`THEN`
			`...`
			`END_IF;`

Word access			`out:= "DW".w0 AND` `"DW".w1;`

Complete the lab using a function instead of coding each network as separate logic.

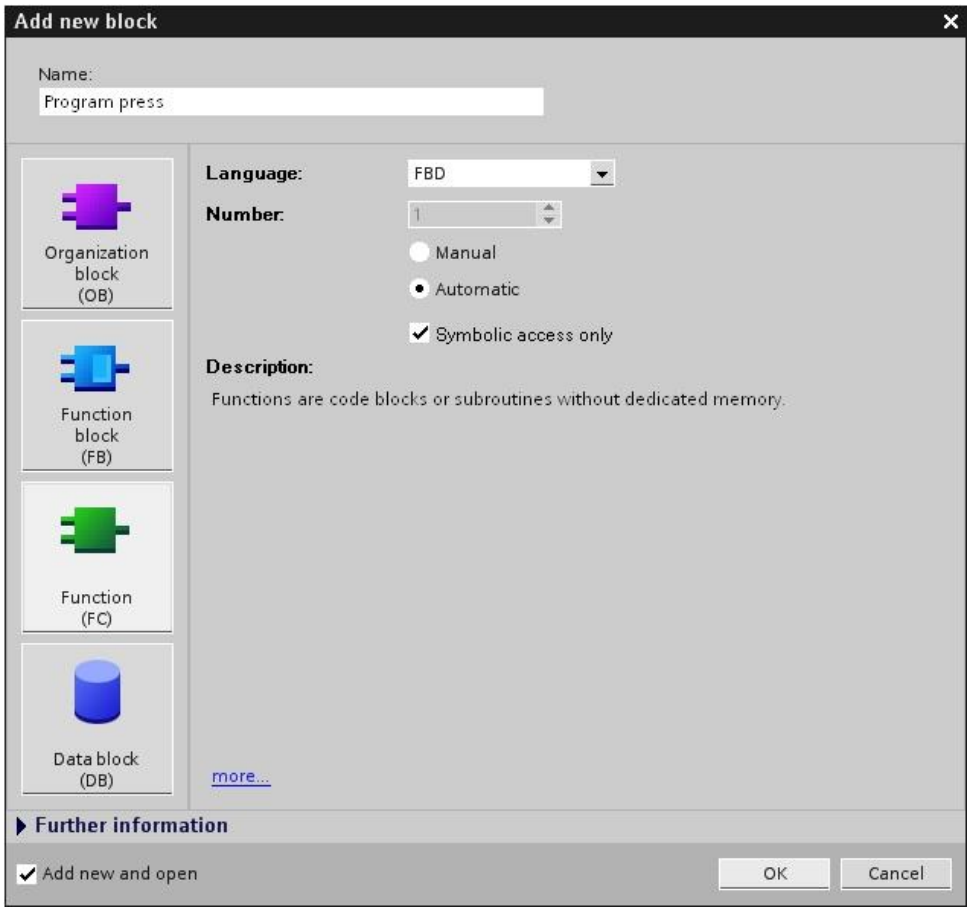
Binary Subtraction:

To perform binary subtraction, the easiest method is to find the 2's complement of the second number and then add the two numbers.

The best method of finding the 2's complement requires the use of a memory bit. The rule requires that bits from the original number be copied to the 2's complement number starting at the right-most bit. The rule applies until a "1" is encountered. The first "1" is copied but a memory bit is set after which the bits are "flipped". Try this rule. It works and may be employed using ladder logic and a Latch bit to quickly find the 2's complement of a number. The logic for finding the 2's complement of a number in ladder logic is begun in Fig. 8-37. Again, logic must be added to complete the function using rungs similar to rungs 4 and 5 of this figure but using bits 2 through15.

Again, code the logic using a function.

To build a function, add a new block in the TIA software:



The block Program Press [FC1] will be opened automatically. However, before the program can be written, the block's interface has to be declared. When the interface is declared, the local variables known only in this block are specified.

The variables consist of two groups

- Block parameters that generate the interface of the block for the call in the program.

Type	Name	Function	Available in
Input parameters	Input	Parameters whose values the block reads	Functions, function blocks and some types of organization blocks
Output parameters	Output	Parameters whose values the block writes	Functions and function blocks
InOut parameters	InOut	Parameters whose value the block reads when it is called and after processing,	Functions and function blocks

	writes again to the same parameter

Local data that is used for storing intermediate results:

Type	Name	Function	Available in
Temporary local data	Temp	Variables that are used for storing temporary intermediate results. Temporary data is retained for one cycle only.	Functions, function blocks and organization blocks
Static local data	Static	Variables that are used for storing static intermediate results in the instance data block. Static data is retained also over several cycles, until it is written anew.	Function blocks

Use a Watch Table to view and change the data. See several examples in videos such as:

S7-1200 Data logging Siemens PLC

Data logging in PLC l Data logging S7 1200 PLC l Tia portal l Siemens