

# Siemens S7-1200

CPU 1212C AC/DC/Relay

## Word Logic Operations

- AND
- OR
- XOR
- INV
- DECODE
- ENCODE
- SELECT
- MUX
- DEMUX



**Code and Compile**  
Learning Made Easy

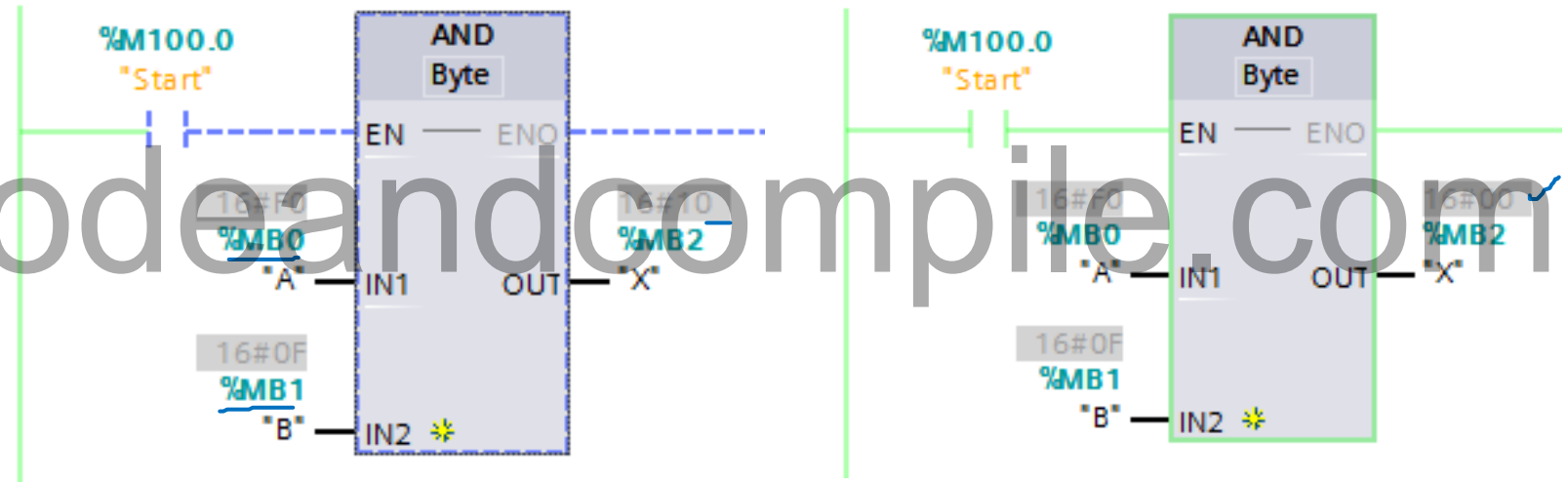
[www.codeandcompile.com](http://www.codeandcompile.com)





## AND Operation

You can use the "**AND logic operation**" instruction to combine the value at the IN1 input and the value at the IN2 input **bit-by-bit by AND logic** and query the result at the OUT output



## How it works?

<b>A</b>	1	1	1	1	0	0	0	0
<b>B</b>	0	0	0	0	1	1	1	1
<b>X</b>	0	0	0	0	0	0	0	0

## Bit wise AND Operation

$$0 . 0 = 0$$

$$0 . 1 = 0$$

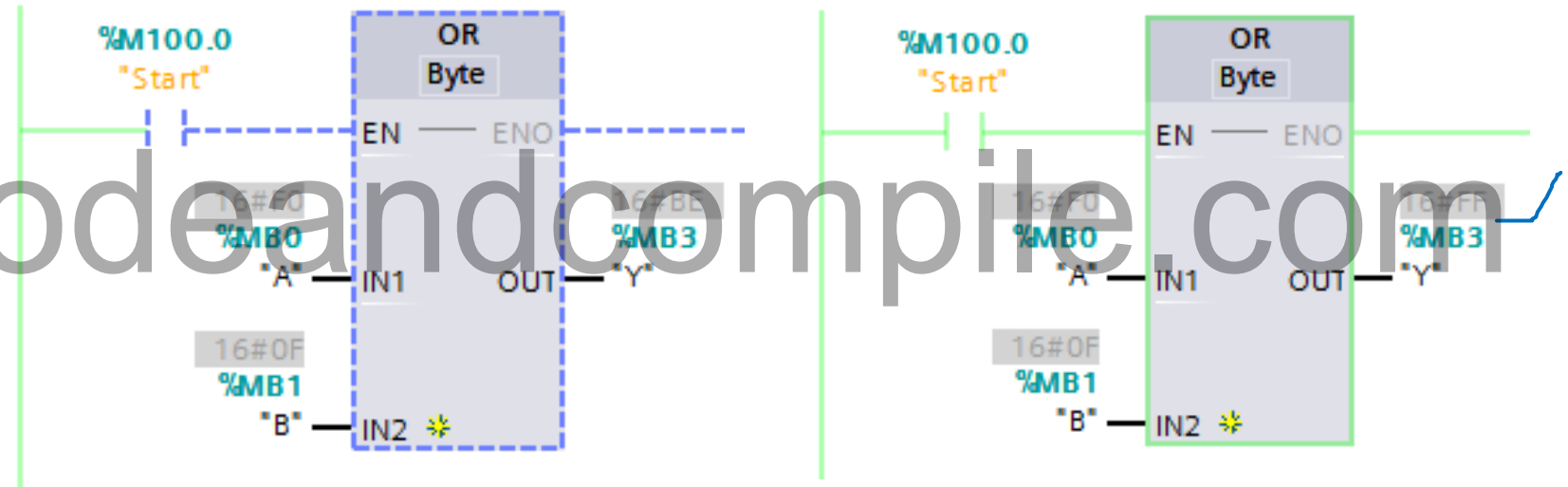
$$1 . 0 = 0$$

$$1 . 1 = 1$$



## OR Operation

You can use the "OR logic operation" instruction to combine the value at the IN1 input and the value at the IN2 input bit-by-bit by OR logic and query the result at the OUT output.



## How it works?

<b>A</b>	1	1	1	1	0	0	0	0
<b>B</b>	0	0	0	0	1	1	1	1
<b>Y</b>	1	1	1	1	1	1	1	1

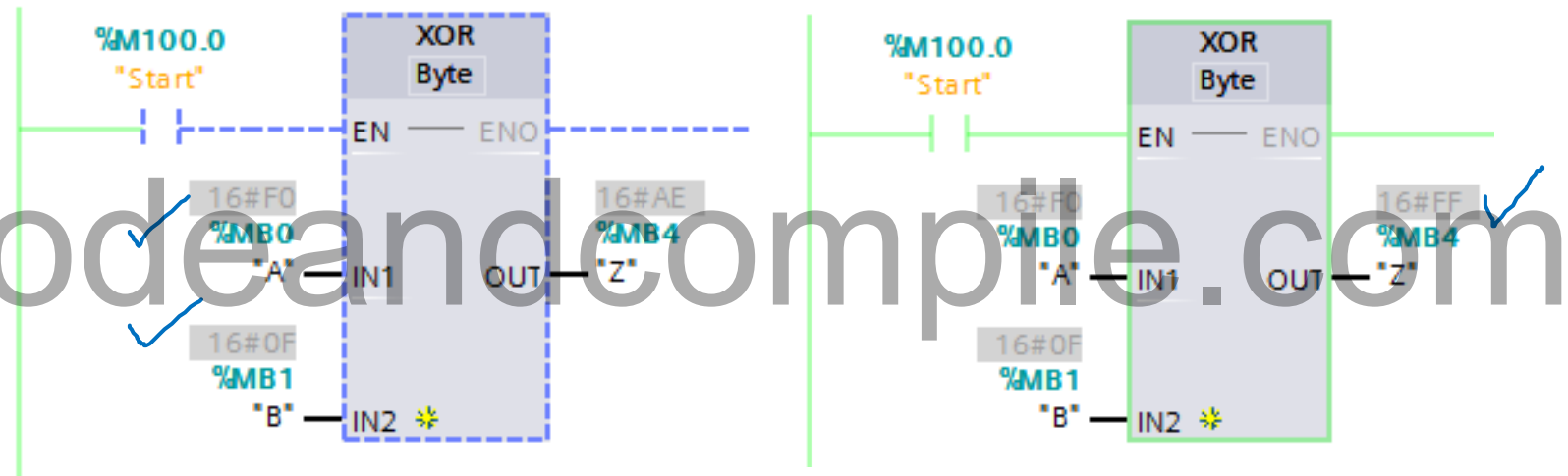
## Bit wise OR Operation

0 . 0 = 0  
0 . 1 = 1  
1 . 0 = 1  
1 . 1 = 1



## XOR Operation

You can use the "EXCLUSIVE OR logic operation" instruction to combine the value at the IN1 input and the value at the IN2 input bit-by-bit by EXCLUSIVE OR logic and query the result at the OUT output.



## How it works?

A	1	1	1	1	0	0	0	0
B	0	0	0	0	1	1	1	1
Z	1	1	1	1	1	1	1	1

## Bit wise XOR Operation

$$0 . 0 = 0$$

$$0 . 1 = 1$$

$$1 . 0 = 1$$

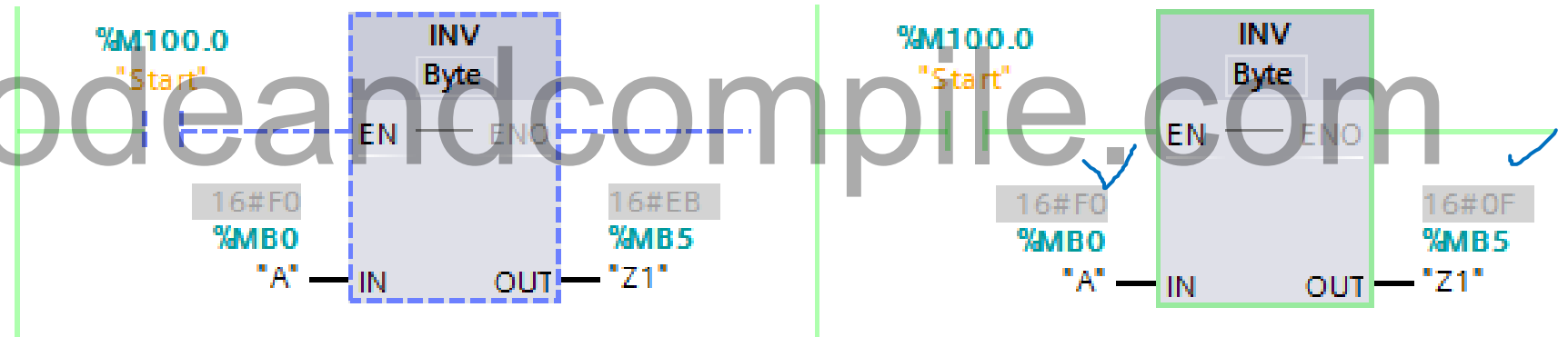
$$1 . 1 = 0$$





## INV Operation

You can use the instruction "Create ones complement" to invert the signal status of the bits at input IN. This inverts the signal state of the individual bits that are then stored at output OUT.



## How it works?

A	1	1	1	1	0	0	0	0
Z1	0	0	0	0	1	1	1	1

## Inverse Operation

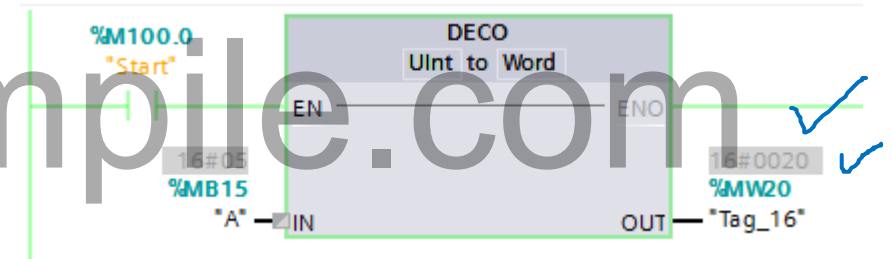
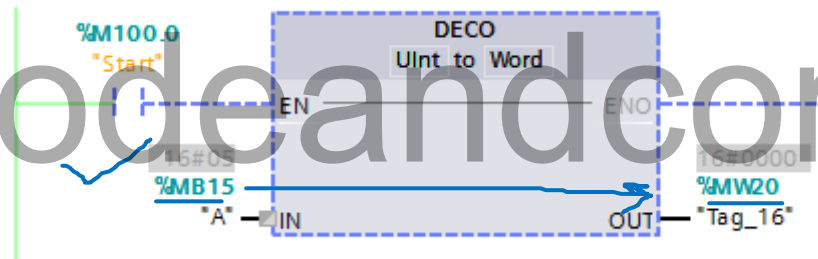
$$\begin{aligned} 0 &= 1 \\ 1 &= 0 \end{aligned}$$



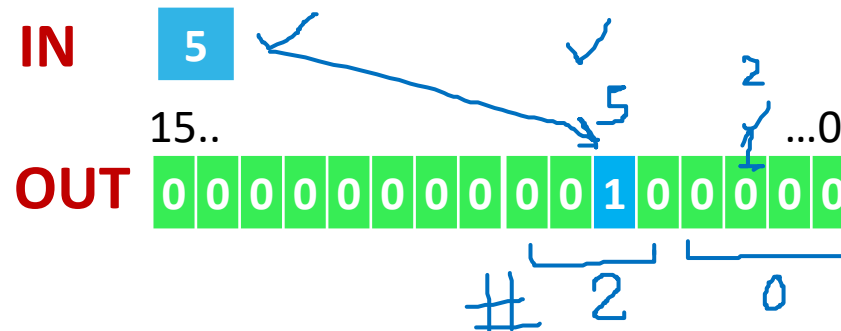
## DECODE Operation

You can use the "Decode" instruction to set a bit in the output value that is specified by the input value.

The "Decode" instruction reads the value at the IN input and sets the bit in the output value whose bit position corresponds to the read value. The other bits in the output value are filled with zeroes.



## How it works?



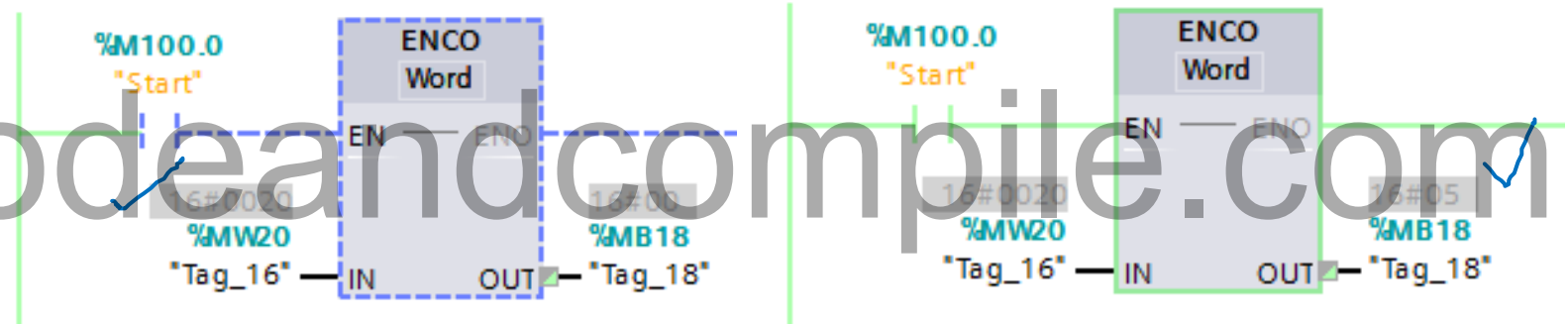
HEX  
= 16#0020



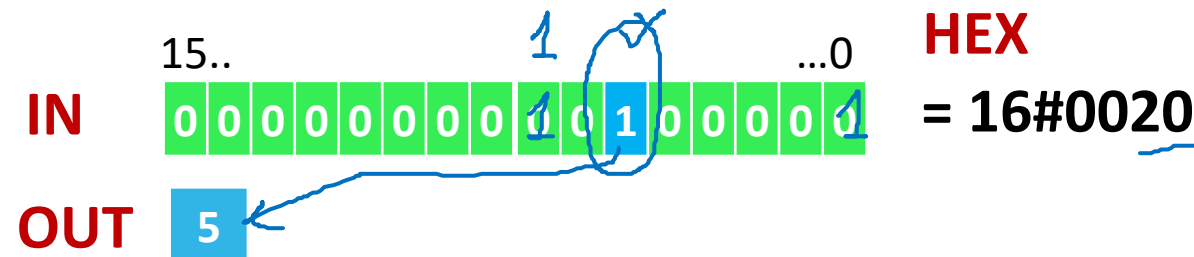
## ENCODE Operation ✓

You can use the "Encode" instruction to read the bit number of the least significant bit in the input value and to send it to the OUT output.

The "Encode" instruction selects the least significant bit of the value at the IN input and writes its bit number to the tag in the OUT output



## How it works?



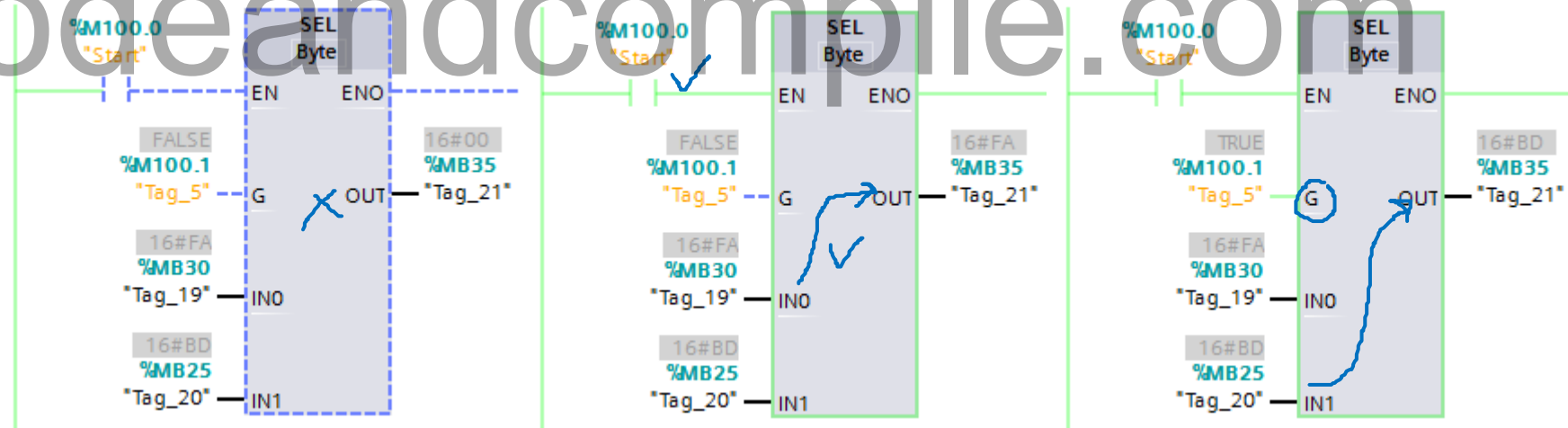


## **SELECT Operation**

Depending on a switch (G input), the "**Select**" instruction selects one of the inputs, IN0 or IN1 and copies its content to the OUT output.

*All tags at all parameters must have the same data type.*

## **How it works?**



**EN = 0 | G = False**  
**OUT = Last State**

**EN = 1 | G = False**  
**OUT = IN0**

**EN = 1 | G = True**  
**OUT = IN1**





## MUX Operation

You can use the instruction "Multiplex" to **copy the content of a selected input** to **output OUT**. The number of selectable inputs of the instruction box can be expanded. *You can declare a maximum of 32 inputs.*

## How it works?

Specifies the input whose content is to be copied.

- If  $K = 0 \Rightarrow$  Parameter **IN0**
- If  $K = 1 \Rightarrow$  Parameter **IN1**, etc

First input value

Second input value

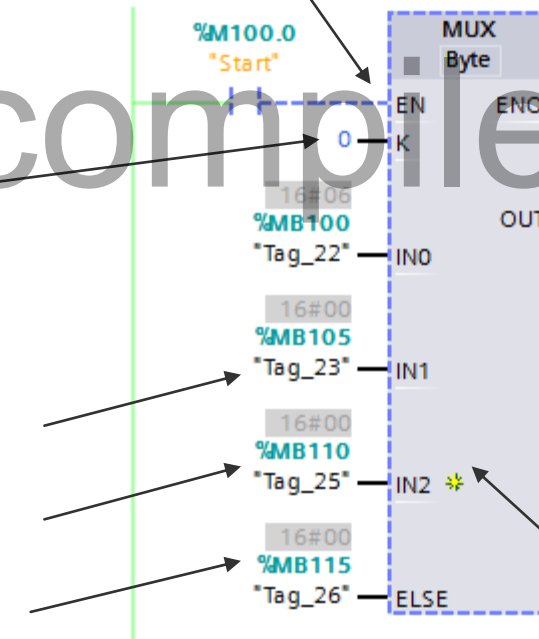
Specifies the value to be copied when  $K > n$ .

Enable Input

Enable Output

Output to which the value is to be copied.

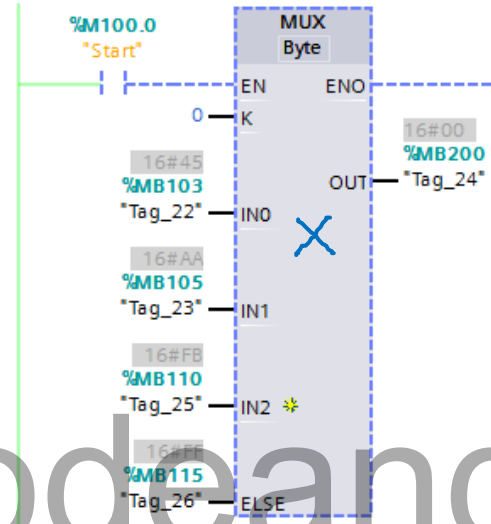
Optional input values



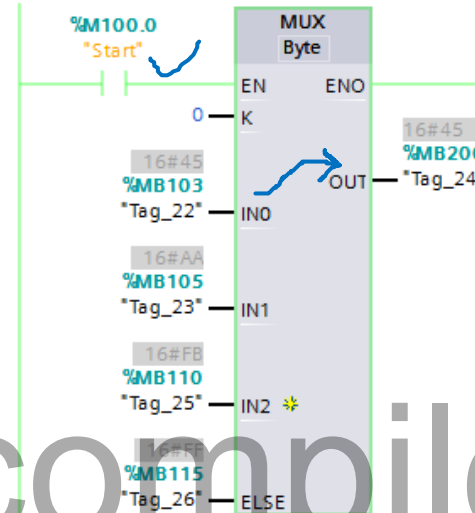
# Understanding Word Operations – MUX - Example



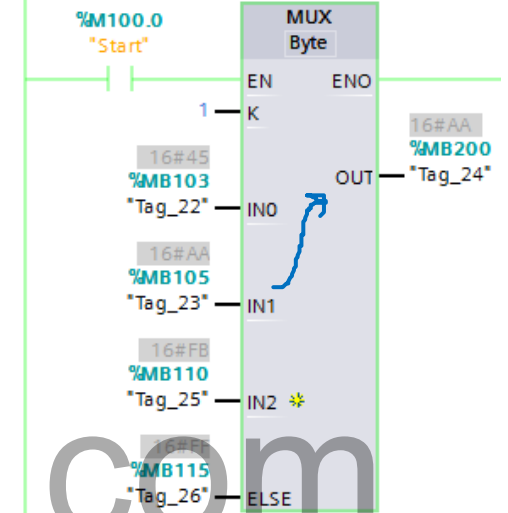
EN = 0 | K = 0 | OUT = Last State



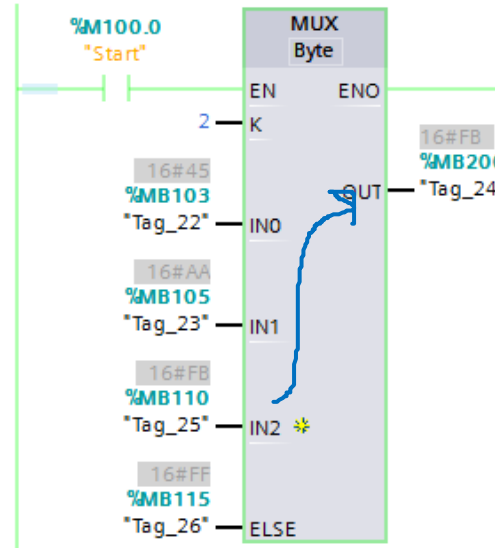
EN = 1 | K = 0 | OUT = IN0



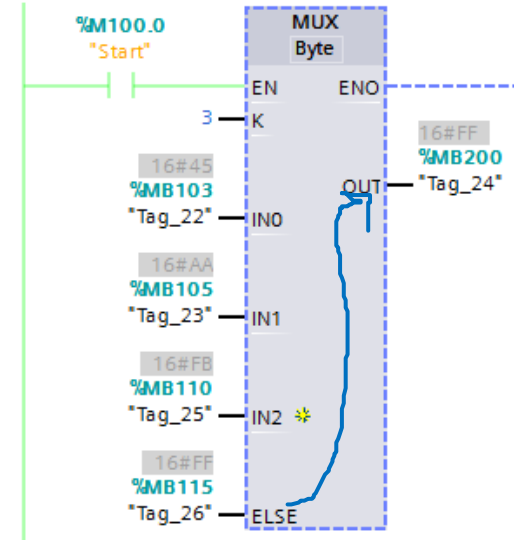
EN = 1 | K = 1 | OUT = IN1



EN = 1 | K = 2 | OUT = IN2



EN = 1 | K = 3 | OUT = ELSE



www.codeandcompile.com



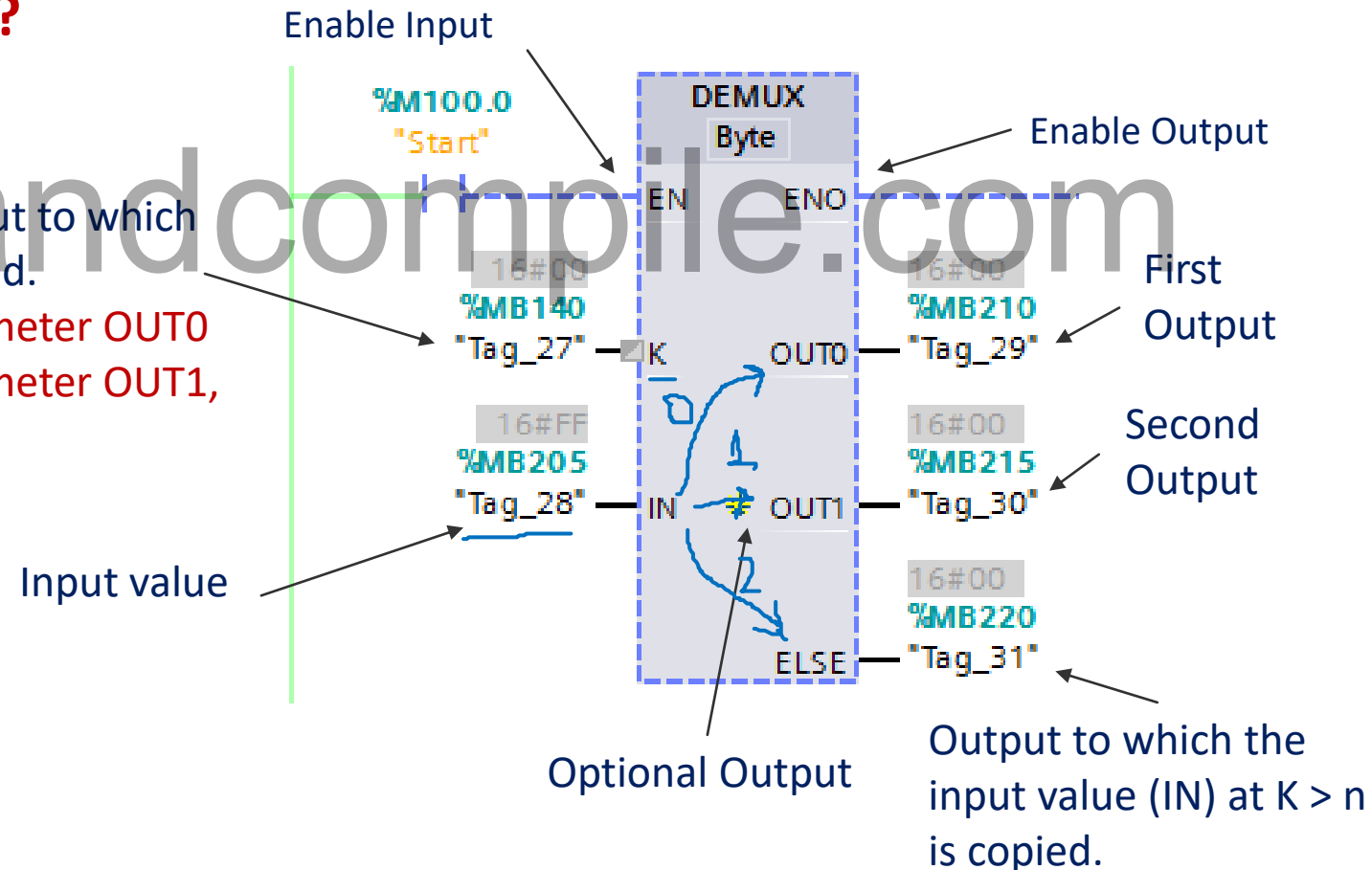
## DEMUX Operation

You can use the instruction "Demultiplex" to **copy the content of the input IN to a selected output**. The number of selectable outputs can be extended in the instruction box. *Numbering starts at OUT0 and continues consecutively with each new output*

## How it works?

Specifies the Output to which input is to be copied.

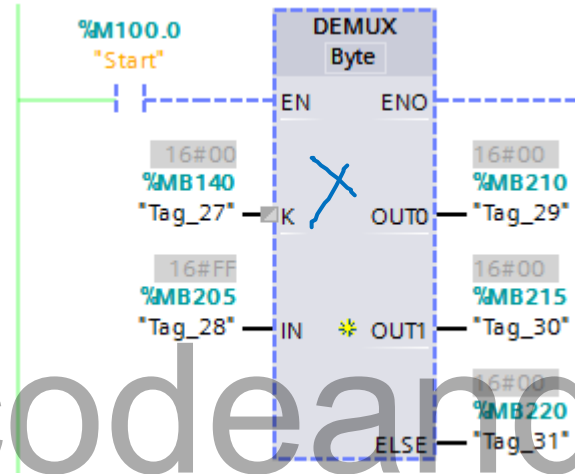
- If  $K = 0 \Rightarrow$  Parameter OUT0
- If  $K = 1 \Rightarrow$  Parameter OUT1, etc.



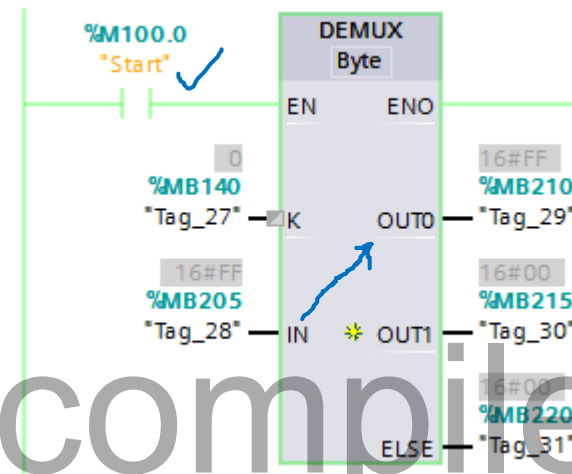
# Understanding Word Operations – DEMUX - Example



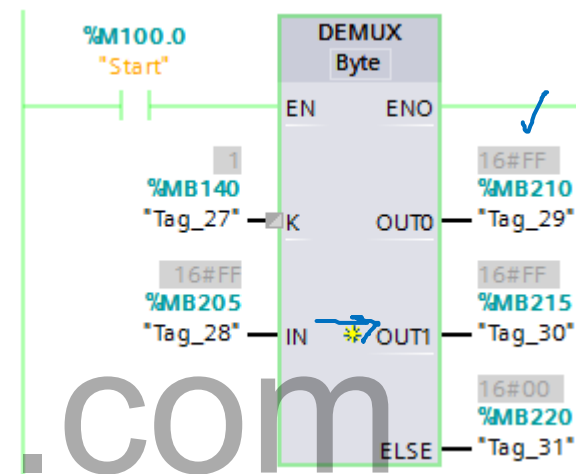
EN = 0 | K = 0  
OUT 0 & 1 = Last State



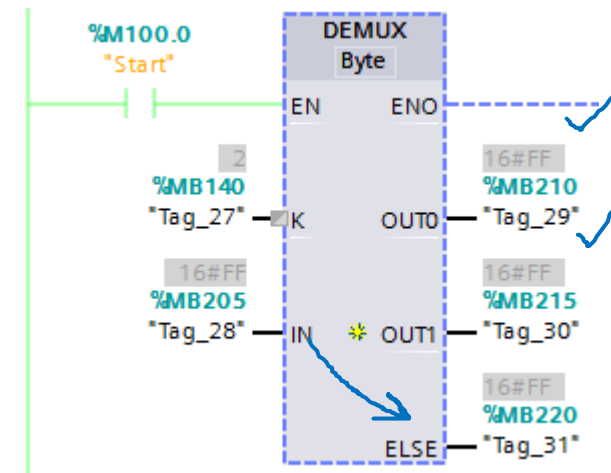
EN = 1 | K = 0  
OUT0 = IN



EN = 1 | K = 1  
OUT1 = IN



EN = 1 | K = 2  
ELSE = IN



www.codeandcompile.com



# What did we learn today?

- **AND:** bitwise AND operation where output is true if both the inputs are TRUE
- **OR:** bitwise OR operation where output is true if any of both the input are TRUE X or
- **XOR:** bitwise OR operation where output is true if any one of the two inputs is TRUE
- **INV:** used for 1's complement of the input
- **DECODE:** instruction to set a bit in the output value that is specified by the input value.
- **ENCODE:** You can use the "Encode" instruction to read the bit number of the least significant bit in the input value and to send it to the OUT output
- **SELECT:** Depending on a switch (G input), the "Select" instruction selects one of the inputs, IN0 or IN1 and copies its content to the OUT output
- **MUX:** You can use the instruction "Multiplex" to copy the content of a selected input to output OUT
- **DEMUX:** You can use the instruction "Demultiplex" to copy the content of the input IN to a selected output.



Code and Compile  
Learning Made Easy

[www.codeandcompile.com](http://www.codeandcompile.com)

## Thank you

*Get copy of this presentation  
in the course!*

