Siemens S7-1200

CPU 1212C AC/DC/Relay

Understanding

Networks, Branches and Rungs

Some Important rules to consider!

Code and Compile
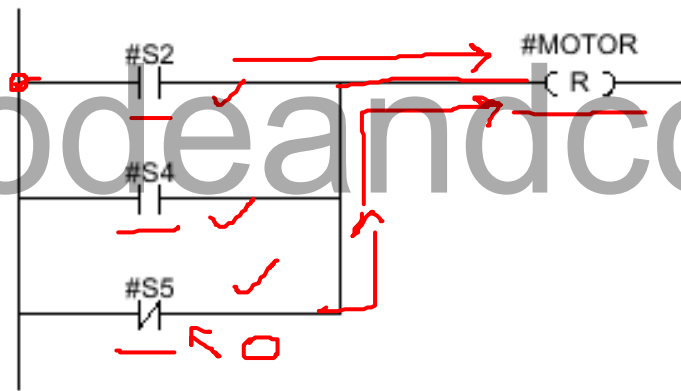Learning Made Easy

www.codeandcompile.com

Branches in Ladder Logic

You use **branches** in main rung to program parallel circuits with the Ladder Logic (LAD) programming language

You can insert several contacts into the branch and thus achieve a parallel circuit of series connections.

The figure below shows an example of the use of branches:

MOTOR carries signal 1, if one of the following conditions is fulfilled:

- S2 is 1
- S4 is 1
- S5 is 0

**The following rules apply to simultaneous branches:**

- Simultaneous **branches are opened downwards** or are connected directly to the power rail. **They are terminated upwards.**
- Simultaneous **branches are opened after the selected LAD element**.
- Simultaneous **branches are terminated after the selected LAD element.**
- **To delete a simultaneous branch**, you must **delete all LAD elements of this branch.**

Crossings in Ladder Logic

A crossing is a place in a LAD network where **one branch is closed** and at the same time **another branch is opened**.

The figure below shows an example of the use of crossing:

Crossing
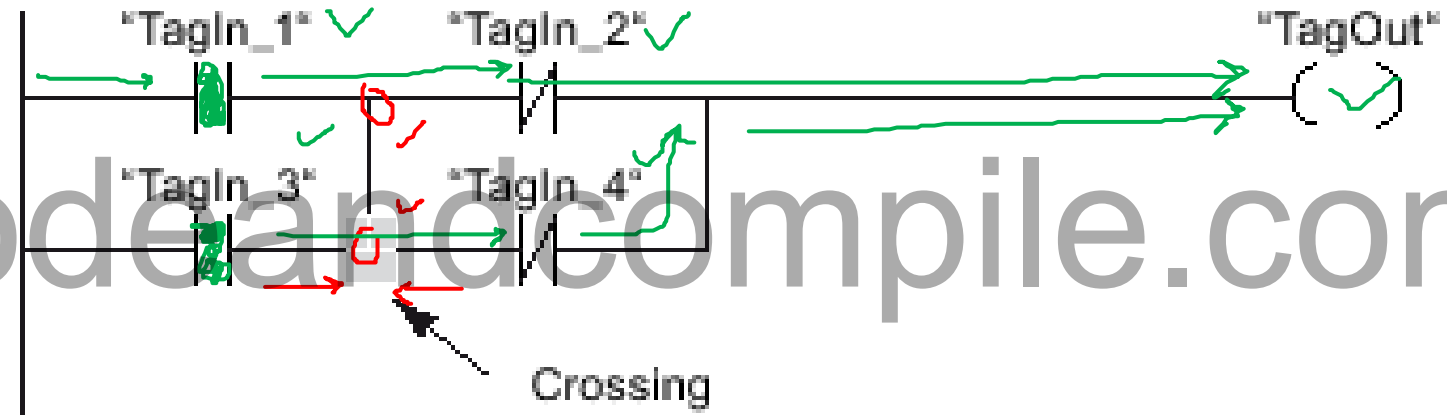
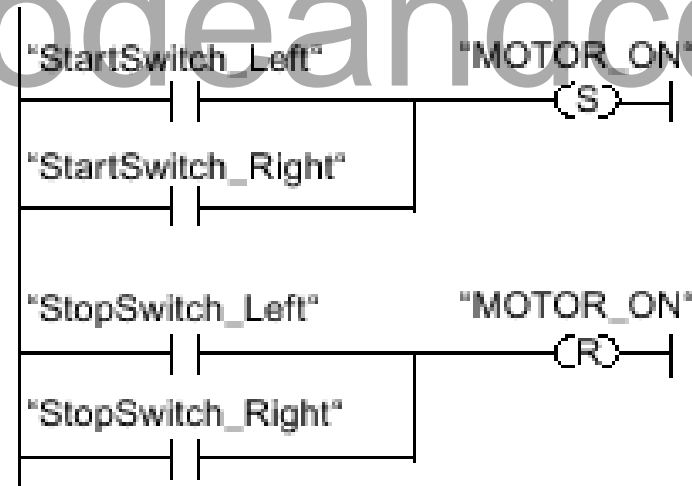**"TagOut" receives signal 1, if the following two conditions are met:**

- "TagIn_1" or "TagIn_3" has signal 1
- "TagIn_2" or "TagIn_4" has signal 0

Rungs in Ladder Logic

The program is mapped in one or more networks. **A network contains a power rail on the left where one or more rungs originate.** The binary signal scans are arranged in the form of contacts on the rungs. **The serial arrangement of the elements on a rung creates a series connection**; arrangement on simultaneous branches creates a parallel connection. A rung is closed by a coil or a box in which the result of logic operation will be written.

The figure below shows an example of the use of several rungs within a network:

```
"StartSwitch_Left"                    "MOTOR_ON"
────┤ ├──────────────────────────────────(S)───┤

"StartSwitch_Right"
────┤ ├────────┘


"StopSwitch_Left"              "MOTOR_ON"
────┤ ├──────────────────────────(R)───┤

"StopSwitch_Right"
────┤ ├────────┘
```

Running rungs

- Rungs and networks are executed from **top to bottom** and from **left to right**.
- This means that **the first instruction in the first rung of the first network is processed first**.
- **All instructions of this rung** are then processed.
- After this come **all other rungs of the first network.**
- The next network is processed only after all rungs have first been run.

Thank you

Get copy of this presentation in the course!

Next Lesson !

Understanding Bit logic Instructions
- NO, NC and OUT

www.codeandcompile.com

Code and Compile
Learning Made Easy

www.codeandcompile.com