# Andhra Pradesh State Skill Development Corporation

Skill AP
APSSDC

eSkill AP
Learn Anytime Anywhere
Andhra Pradesh State Skill Development Corporation


# Basics of PLC

## Flip flops, Positive and Negative edge detection with an example

# Flip-Flops

Flip Flop

A flip flop has a Set input and a Reset input. The memory bit is set or reset, depending on which input has an RLO=1.

Priority

If there is an RLO=1 at both inputs at the same time, the priority must be determined. In LAD and FBD there are different symbols for Dominant Set and Dominant Reset memory functions. In STL, the instruction that was programmed last has priority.

Note

With a warm restart of the CPU, all outputs are reset. That is, they are overwritten with the state ‘0’.

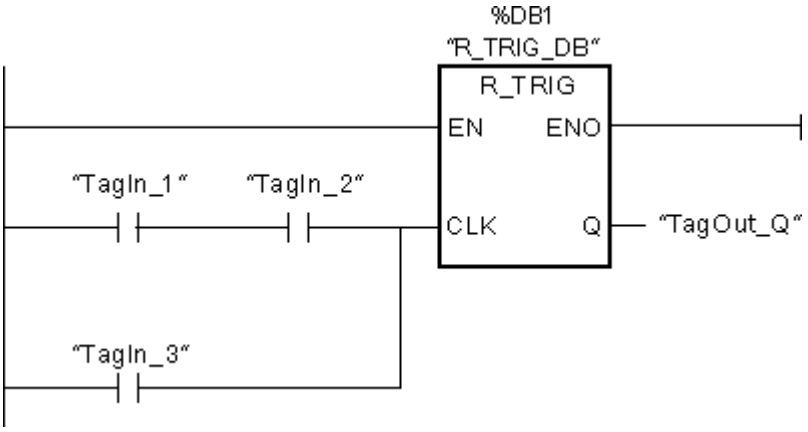
## Signal-Edge Detection

Signal edge detection (P, N)

With a signal edge detection it is possible to detect the status change of an individual operand (in the example "T_ON") from '0' to '1' (rising or positive edge) or from '1' to '0' (falling or negative edge). If this is the case, the instruction supplies RLO '1' as the result, which can be further logically linked (in the example as set condition) or assigned to another operand (for example, a memory bit) as status. In the following cycle, the instruction then once again supplies '0' as the result even if "T_ON" still is status '1'.

The instruction compares the current status of the operand "T_ON" with its status in the previous program cycle. This status is stored in a so-called edge memory bit for this (in the example "M_Fl_ON"). It must be ensured that the status of this edge memory bit is not overwritten at another location in the program. For each edge detection, a separate edge memory bit must be used accordingly, even then when the same operand (in the example, "T_ON") is detected once again, for example, in another block!

The following example shows how the instruction works:



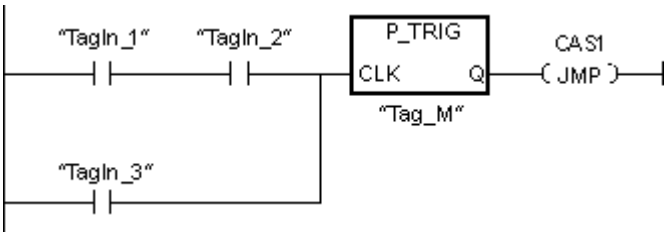

The previous state of the tag at the CLK input is stored in the "R_TRIG_DB" tag. If a change in the signal state from "0" to "1" is detected in the "TagIn_1" and "TagIn_2" operands or in the "TagIn_3" operand, the "TagOut_Q" output has signal state "1" for one cycle.

## RLO-Edge Detection

RLO edge detection (P=, P_TRIG)

An RLO edge detection detects whether the status of an individual operand or the RLO of a logic operation has changed from '0' to '1' (rising or positive edge) or from '1' to '0' (falling or negative edge). If this is the case, both edge detections supply RLO '1' as a result to their output for the duration of one cycle. In the following cycle, the instructions then once again supply RLO '0' as a result, even if the status or the RLO of the operand or the logic operation has notchanged. The instructions compare the current status of the operand or the RLO of the logic operation with its status in the previous program cycle which is stored in a so-called edge memory bit for this (in the example, "M_Fl_Count_pos" or "M_Fl_Count_neg). It must be ensured that the status of this edge memory bit is not overwritten at another location in the program. For every edge detection, a separate edge memory bit must be used accordingly, even then when the same operand is detected once again, for example, in another block!

The following example shows how the instruction works:




The RLO of the previous query is saved in the edge memory bit "Tag_M". If a "0" to "1" change is detected in the signal state of the RLO, the program jumps to jump label CAS1.