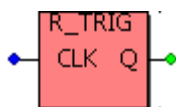


1.2.1.3. Стандартні функціональні блоки

Детектори імпульсів

Детектори імпульсів застосовуються, коли має бути реакція не на стан дискретного сигналу, а на його зміну.

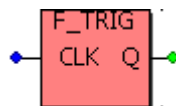
Детектор переднього фронту *R_TRIG*



CLK	BOOL	будь-яка булева змінна.
Q	BOOL	TRUE, коли CLK змінюється з FALS на TRUE. У решті випадків – FALS.

Опис: Генерує одиничний імпульс по передньому фронту вхідного сигналу, тобто вихід Q установлюється в TRUE, коли у попередньому циклі вхід CLK дорівнював FALS, а у поточному циклі він вже має значення TRUE.

Детектор заднього фронту *F_TRIG*



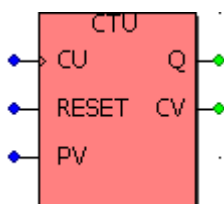
CLK	BOOL	будь-яка булева змінна.
Q	BOOL	TRUE, коли CLK змінюється з TRUE на FALS. У решті випадків – FALS.

Опис: Генерує одиничний імпульс по задньому фронту вхідного сигналу, тобто вихід Q установлюється в TRUE, коли у попередньому циклі вхід CLK дорівнював TRUE, а у поточному циклі він вже має значення FALS.

Попередження: Блок *F_TRIG* має властивість формувати хибний імпульс при перезапуску.

Лічильники

Інкрементний лічильник *CTU*

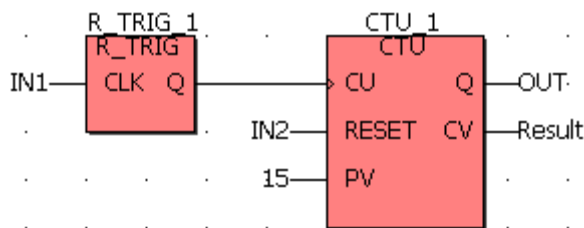


CU	BOOL	– вхід для лічіння (лічити, коли CU дорівнює TRUE).
RESET	BOOL	– команда скидання (домінанта).
PV	DINT	- програмоване максимальне значення.
Q	BOOL	- переповнення: TRUE, коли CV >= PV.
CV	DINT	- результат лічіння.

Опис: По кожному фронту на CU-вході значення лічильника на CV-виході збільшується на 1. Вихід Q установлюється у TRUE, коли лічильник досягає або перевищує задане значення на PV-вході. Логічна одиниця на RESET-вході зупиняє лічильник та скидає його показання (CV:=0).

Попередження: блок CTU не визначає передній або задній фронт входу CU. Для того, щоб створити імпульсний лічильник, його потрібно сполучити з блоками R_TRIG або F_TRIG.

Приклад:



(*ST еквівалент: R_TRIG_1- це екземпляр блока R_TRIG, а CTU_1- це екземпляр блоку CTU*)

CTU_1(R_TRIG_1(IN1),IN2,15);

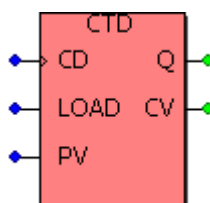
OUT:=CTU_1.Q;

Result:=CTU_1.CV;

(*IL еквівалент*)

<i>LD</i>	<i>IN1</i>	<i>LD</i>	<i>15</i>
<i>ST</i>	<i>R_TRIG_1.CLR</i>	<i>ST</i>	<i>CTU_1.PV</i>
<i>CAL</i>	<i>R_TRIG_1</i>	<i>CAL</i>	<i>CTU_1</i>
<i>LD</i>	<i>R_TRIG_1.Q</i>	<i>LD</i>	<i>CTU_1.CU</i>
<i>ST</i>	<i>CTU_1.CU</i>	<i>ST</i>	<i>OUT</i>
<i>LD</i>	<i>IN2</i>	<i>LD</i>	<i>CTU_1.CV</i>
<i>ST</i>	<i>CTU_1.Reset</i>	<i>ST</i>	<i>Result</i>

Декрементний лічильник CTD



CD BOOL – вхід для лічіння (лічити, коли CD дорівнює TRUE).

LOAD BOOL – команда завантаження (домінанта) .

PV DINT - програмоване максимальне значення.

Q BOOL - переповнення: TRUE, коли CV=0.

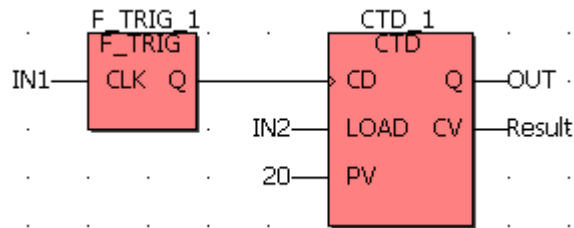
CV DINT - результат лічіння.

Опис: По кожному фронті на CD-вході значення лічильника на CV-виході зменшується на 1. Вихід Q устанавлюється у TRUE, коли лічильник досягає нуля. CV-лічильник завантажується початковим значенням, яке дорівнює PV, коли вхід LOAD= TRUE.

Попередження: блок CTD не визначає передній і задній фронт входу CD. Для того, щоб створити імпульсний лічильник, його потрібно сполучити з блоками R_TRIG або F_TRIG.

Приклад:

(*Програма FBD, що використовує блок CTD*)



(* ST еквівалент: F_TRIG_1- це екземпляр блоку F_TRIG, а CTD_1- це екземпляр блоку CTD*)

CTD(F_TRIG(IN1),IN2,20);

OUT:=CTD_1.Q;

Result:=CTD_1.CV

(*IL еквівалент*)

LD IN1

ST F_TRIG_1.CLR

CAL F_TRIG_1

LD F_TRIG.Q

ST CTD_1.CD

LD IN2

ST CTD_1.LOAD

LD 20

ST CTD_1.PV

CAL CTD_1

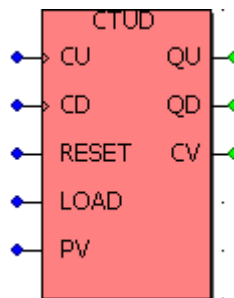
LD CTD_1.Q

ST OUT

LD CTD_1.CV

ST Result

Інкрементний-декрементний лічильник CTUD



CU BOOL – вхід для лічіння угору (лічити, коли CU дорівнює TRUE).

CD BOOL – вхід для лічіння униз (лічити, коли CD дорівнює TRUE).

RESET BOOL – команда скидання (домінанта). (CV=0, коли RESET дорівнює TRUE).

LOAD BOOL – команда завантаження (CV=PV, коли LOAD дорівнює TRUE).

PV DINT - програмоване максимальне значення.

QU BOOL - переповнення зверху. TRUE, коли CV>=PV.

QD BOOL - переповнення знизу. TRUE, коли CV=0.

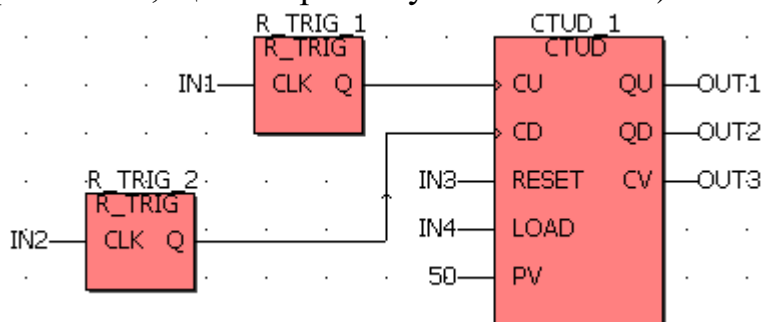
CV DINT - результат лічіння.

Опис: По фронту на CU-вході лічильник збільшується на 1. По фронту на CD-вході лічильник зменшується на 1 до 0. Коли RESET- вхід дорівнює TRUE значення CV-виходу скидається у 0. Коли LOAD-вхід дорівнює TRUE CV-вихід завантажується значенням, що дорівнює PV- входу.

Попередження: блок CTUD не визначає передній або задній фронт входів лічильника (CU I CD). Для створення імпульсного лічильника його потрібно сполучити з блоками R_TRIG або F_TRIG.

Приклад:

(* Програма FBD, що використовує блок CTUD*)



(* ST еквівалент: R_TRIG_1 і R_TRIG_2 – це екземпляри блока R_TRIG, а CTUD_1 – це екземпляр блока CTUD*).

CTUD_1 (*R_TRIG_1*(*IN1*), *R_TRIG_2*(*IN2*), *IN3*,*IN4*,50);

OUT1:=CTUD_1.QU;

OUT2:=CTUD_1.QD;

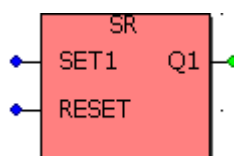
OUT3:=CTUD_1.CV;

(*IL еквівалент *)

<i>LD IN1</i>	<i>LD IN4</i>
<i>ST R_TRIG_1.CLK</i>	<i>ST CTUD_1.LOAD</i>
<i>CAL R_TRIG_1</i>	<i>LD 50</i>
<i>LD R_TRIG_1.Q</i>	<i>ST CTUD_1.PV</i>
<i>ST CTUD_1.CU</i>	<i>CAL CTUD_1</i>
<i>LD IN2</i>	<i>LD CTUD_1.QU</i>
<i>ST R_TRIG_2.CLK</i>	<i>ST OUT1</i>
<i>CAL R_TRIG_2</i>	<i>LD CTUD_1.QD</i>
<i>LD R_TRIG_2.Q</i>	<i>ST OUT2</i>
<i>ST CTUD_1.CD</i>	<i>LD CTUD_1.CV</i>
<i>LD IN3</i>	<i>ST OUT3</i>
<i>ST CTUD_1.RESET</i>	

Тригери

Перемикач із домінантою вмикання SR



SET1 BOOL - якщо TRUE, встановлює Q1 у TRUE (домінанта).

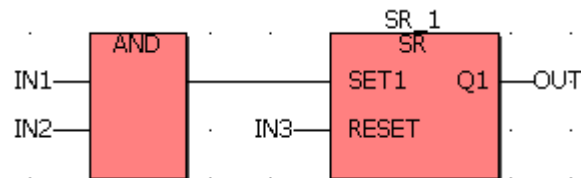
RESET BOOL - якщо TRUE, скидає Q1 у FALS.

Q1 BOOL - логічний стан пам'яті.

Опис: Блок SR має на Q1-виході два стійких стана – TRUE і FALS. SET1-вхід вмикає вихід, а RESET- вхід вимикає його. При одночасному впливі обох входів SET1-вхід є домінантним.

Set1	Reset	Q1	result Q1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Приклад:
 (*FBD-програма , що використовує блок SR*)



(*ST - еквівалент: SR_1- це екземпляр блока SR*)

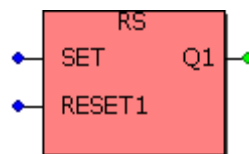
SR_1((IN1 AND IN2),IN3);

OUT:=SR_1.Q1;

(* IL – еквівалент*)

<i>LD IN1</i>	<i>ST SR_1.RESET</i>
<i>AND IN2</i>	<i>CAL SR_1</i>
<i>ST SR_1.SET1</i>	<i>LD SR_1.Q1</i>
<i>LD IN3</i>	<i>ST OUT</i>

Перемикач із доміантою вимкнення RS



SET BOOL - якщо TRUE встановлює Q1 у TRUE .

RESET1 BOOL - якщо TRUE, скидає Q1 у FALS (домінанта).

Q1 BOOL - логічний стан пам'яті.

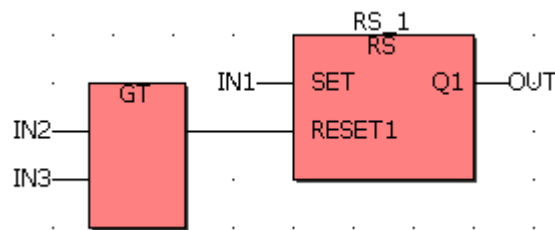
Опис: Блок RS має на Q1-виході два стійких стана – TRUE і FALS. SET-вхід вмикає вихід, а RESET1- вхід вимикає його. При одночасній дії обох входів RESET1-вхід є доміантним.

Set	Reset1	Q1	result Q1
0	0	0	0

0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Приклад:

(*FBD-програма , що використовує блок RS*)



(*ST - еквівалент: RS_1- це екземпляр блока RS*)

RS_1(*IN1*, (*IN2* *GT* *IN3*));

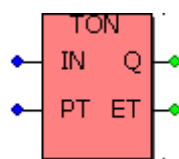
OUT:=RS_1.Q1;

(* IL – еквівалент*)

```
LD  IN1          ST  RS_1.RESET1
ST  RS_1.SET     CAL  RS_1
LD  IN2          LD  RS_1.Q1
GT  IN3          ST  OUT
```

Таймери

Таймер із затримкою вмикання TON



IN **BOOL** - якщо передній фронт, то починає збільшувати внутрішній таймер, якщо задній фронт, то зупиняється і скидає внутрішній таймер.

PT **TIME** - максимальний програмований час.

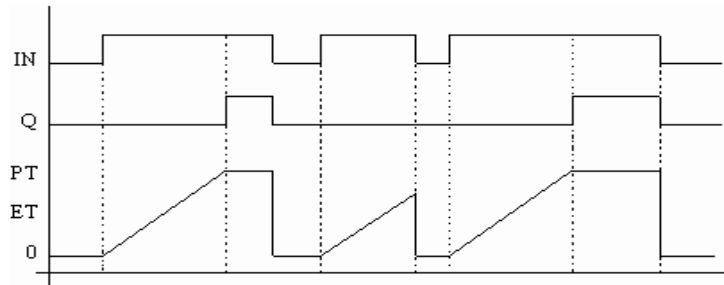
Q **BOOL** - якщо TRUE, програмований час використаний.

ET **TIME** - поточний використаний час.

Опис: По передньому фронту **IN**- входу виконується скид лічильника **ET** і починається новий відлік часу. **Q**- вихід встановлюється у **TRUE** через заданий на **PT**-вході час, якщо **IN**-вхід продовжує залишатися у стані **TRUE**. По задньому фронту **IN**- входу зупиняється відлік часу і **Q**-вихід скидається у **FALS**. Таким чином, **Q**- вихід вмикається логічною одиницею, яка триває

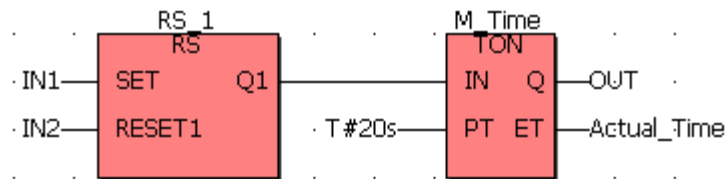
протягом часу показаному у PT – вході, а вимикається по задньому фронті IN- входу.

Часова діаграма має вигляд:



Приклад:

(*FBD-програма , що використовує блок TON*)



(*ST - еквівалент: RS_1- це екземпляр блока RS, а M_Time - екземпляр блока TON*)

$M_Time(RS_1(IN1,IN2),T\#20s)$

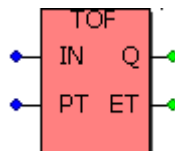
$OUT:=M_TIME_1.Q;$

$Actual_Time:=M_TIME_1.ET;$

(* IL – еквівалент*)

$LD\ IN1$	$LD\ T\#20s$
$ST\ RS_1.SET$	$ST\ M_Time.PT$
$LD\ IN2$	$CAL\ M_Time$
$ST\ RS_1.RESET1$	$LD\ M_Time.Q$
$CAL\ RS_1$	$ST\ OUT$
$LD\ RS_1.Q1$	$LD\ M_Time.ET$
$ST\ M_Time.IN$	$ST\ Actual_Time$

Таймер із затримкою вимикання TOF



IN BOOL - якщо задній фронт, то починає збільшувати внутрішній таймер, якщо передній фронт, то зупиняється і скидає внутрішній таймер.

PT TIME - максимальний програмований час.

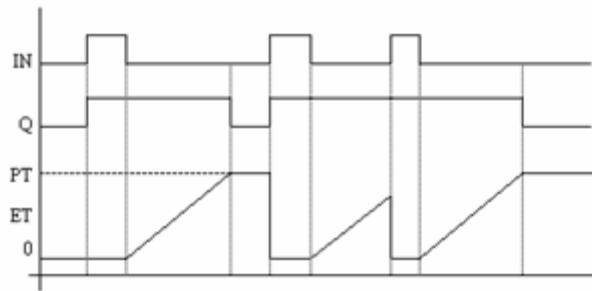
Q BOOL - якщо TRUE, програмований час використаний.

ET TIME - поточний використаний час.

Опис: По задньому фронті IN- входу виконується скид лічильника ET, Q- вихід устанавлюється у TRUE і починається новий відлік часу. Через заданий на PT- вході час Q- вихід скидається. Якщо під час відліку IN- вхід

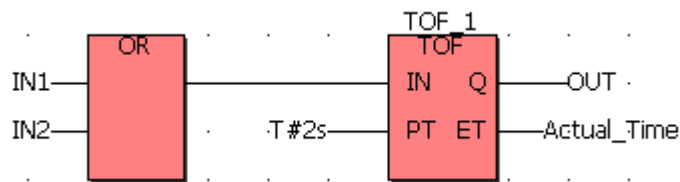
установлюється у стані TRUE, то відлік призупиняється. Таким чином, Q-вихід вмикається по задньому фронту, а вимикається логічним нулем тривалістю не менш, як на PT-вході.

Часова діаграма має вигляд:



Приклад:

(*FBD-програма , що використовує блок TOF*)



(*ST - еквівалент: TOF_1- екземпляр блока TOF*)

TOF_1((IN1 OR IN2), T#2s);

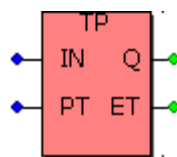
OUT:= TOF_1.Q;

Actual_Time:=TOF_1.ET

(* IL – еквівалент*)

<i>LD IN1</i>	<i>CAL TOF_1</i>
<i>OR IN2</i>	<i>LD TOF_1.Q1</i>
<i>ST TOF_1.IN</i>	<i>ST OUT</i>
<i>LD T#2s</i>	<i>LD TOF_1.ET</i>
<i>ST TOF_1.PT</i>	<i>ST Actual_Time</i>

Час пульсування TP



IN BOOL - якщо передній фронт, то починає збільшувати внутрішній таймер (якщо вже не збільшується).

Якщо FALSE і лише, якщо таймерний інтервал вичерпано, скидає внутрішній таймер.

Будь-яка змінна IN під час лічіння не має ефекту.

PT TIME - максимальний програмований час.

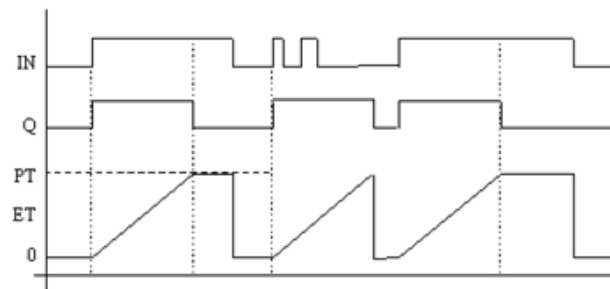
Q BOOL - якщо TRUE таймер лічить.

ET TIME поточний використаний час.

Опис: Запуск таймера здійснюється по передньому фронту імпульсу на IN-вході, PT-вхід задає тривалість імпульсу на Q-виході . Після запуску таймер не реагує на зміну значення IN- входу. ET-вихід лічить час і, коли

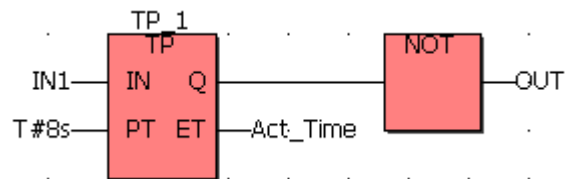
значення досягає установленного на РТ-вході, лічильник зупиняється і Q-вихід скидається у 0.

Часова діаграма має вигляд:



Приклад:

(*FBD-програма , що використовує блок TP*)



(*ST - еквівалент: TP_1- екземпляр блока TP*)

TP_1(IN1,T#8s);

Act_Time:=TP_1.ET;

OUT:=NOT(TP_1.Q);

(* IL – еквівалент*)

<i>LD IN1</i>	<i>LD TP_1.Q</i>
<i>ST TP_1.IN</i>	<i>NOT</i>
<i>LD T#8s</i>	<i>ST OUT</i>
<i>ST TP_1.PT</i>	<i>LD TP_1.ET</i>
<i>CAL TP_1</i>	<i>ST Act_Time</i>