FAQ-01X4QJPN • 02/2016

Description of Free Function Blocks

SINAMICS G120 / G120P modular and SINAMICS G120C

https://support.industry.siemens.com/cs/ww/en/view/85168215

This entry is from the Siemens Industry Online Support. The general terms of use (http://www.siemens.com/terms_of_use) apply.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens' products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit http://support.industry.siemens.com.

Table of contents

1 Overview

1.1 Application Range, Characteristics

A logic operation, which combines several states (e.g. access control, plant status) to a control signal (e.g. ON command), is required for controlling the drive system in a wide variety of applications. Along with logic operations, a number of mathematical functions and latching elements are becoming increasingly important in drive systems.

This functionality is available as a function module "Free function blocks" (FFB) within the:

- SINAMICS G120 for the control units CU240B-2, CU240E-2, and CU250S-2,
- SINAMICS G120 / G120P for the control unit CU230P-2
- SINAMICS G120C

1.2 Configuration and Operation

The free function blocks are configured at the parameter level. The following parameters are required for this:

- Input parameters (e.g. inputs I0 ... I3 for the AND function block).
- Output parameters (e.g. output Y for the numerical change-over switch).
- Adjustable parameters (e.g. pulse duration for pulse generator MFP).
- Runtime group (this includes the sampling time; the free function blocks are not computed in the factory setting).
- Run sequence within the runtime group.

A parameter is assigned to each input, output, and adjustable variable. These can be accessed by means of the Basic Operator Panel (BOP-2), the Intelligent Operator Panel (IOP) or the STARTER commissioning tool. The "free function blocks" can all be interconnected at the BICO level.

The "free function blocks" do not support data set dependency.

Note Within the control unit CU250S-2 the free function blocks are combined into a function module, which has to be released by parameter p0108[1].18 during quick commissioning. Within the STARTER the release has to be set by the wizzard of the quick commissioning.

1.3 Range of Free Function Blocks

The table below shows the range of free function blocks available. For information about the special technical properties of the individual function blocks, see the function diagrams in Section 2.

Code	Name of the function block	Data type	No. per control unit
AND	AND	BOOL	4
OR	OR	BOOL	4
XOR	XOR	BOOL	4
NOT	Inverter	BOOL	6
ADD	Adder	REAL	3
SUB	Subtractor	REAL	2
MUL	Multiplier	REAL	2
DIV	Divider	REAL	2
AVA	Absolute value generator with sign evaluation	REAL	2
NCM	Numeric comparator	REAL	2
PLI	Scaling polyline	REAL	2
MFP	Pulse generator	BOOL	2
PCL	Pulse contractor	BOOL	2
PDE	ON delay device	BOOL	2
PDF	OFF delay device	BOOL	2
PST	Pulse stretcher	BOOL	2
RSR	RS flip flop, reset dominant	BOOL	2
DFR	D flip flop, reset dominant	BOOL	2
BSW	Binary change-over switch	BOOL	2
NSW	Numeric change-over switch	REAL	2
LIM	Limiter	REAL	2
PT1	Smoothing element	REAL	2
INT	Integrator	REAL	1
DIF	Derivative action element	REAL	1
LVM	Double-sided limit monitor with hysteresis	BOOL	2

Table 1: List of the existing free function blocks

1.4 Runtime Group, Sampling Time and Run Sequence

Runtime groups

Runtime groups are groups of free function blocks within the system that are computed in the same sampling time and at a specific time. A total of 6 runtime groups are available. Each function block is assigned one runtime group via a parameter. In the factory setting, the value 9999 (i.e. the function block is not computed) is assigned to each function block.

Sampling time

For the control units of the SINAMICS G120 / G120P and SINAMICS G120C the sampling times Ts are preset to fixed values. It is defined for each free function bloc, in which runtime groups/sampling times it can be computed.

	Run-time group					
	1	2	3	4	5	6
	r20001[1] = 8 ms	r20001[2] = 16 ms	r20001[3] = 32 ms	r20001[4] = 64 ms	r20001[5] = 128 ms	r20001[6] = 256 ms
Logic function blocks AND, OR, XOR, NOT	X	X	X	X	X	X
Arithmetic function blocks ADD, SUB, MUL, DIV, AVA, NCM, PLI	-	-	-	-	X	X
Time function blocks MFP, PCL, PDE, PDF, PST	-	-	-	-	X	X
Memory function blocks RSR, DSR	X	X	X	X	X	X
Switch function block NSW	-	-	-	-	X	X
Switch function block BSW	X	X	X	X	X	X
Control function blocks LIM, PT1, INT, DIF	-	-	-	-	X	X
Complex function blocks LVM	-	-	-	-	X	X

Table 2: Sample times of the free function blocks

Run Sequence

In the factory setting, each free function block is assigned a default setting for the run sequence. A function block with a lower run sequence value is computed within a runtime group before one with a higher value. The run sequence of consecutive free function blocks within a runtime group can be optimized by changing these values accordingly.

A run sequence value can be used within a runtime group once only. The run sequence can be set to between 0 and 32000.

Note If a configuration is carried out OFFLINE, each run sequence value can be initially set at the beginning (e.g. a value can also be assigned to more than one function block simultaneously). The system does not check this until the configuration has been downloaded to the Control Unit.

Note Once downloaded, the parameter values are checked in the order of the parameter numbers. If the system detects that the run sequence value for one function block is already being used by a different function block, the value is not applied and fault F01042 (message in STARTER: Error occurred when downloading) is output.

Note In the factory setting, the value range 10 ... 750 is already assigned the run sequence values of the function blocks. In user configurations, for example, the only run sequence values outside this range which should be used are those above 1000. This will avoid conflicts during the download process with the run sequence values that have already been assigned.

1.5 Connection to the Drives Parameter

Connector inputs (CI) and connector outputs (CO) on the free function blocks (p20094 ... p20381) have the properties of per unit variables. This means that calculations in the free function blocks are only carried out with per unit signal values (1.0 = 100%). Conversion to the connectors of the drive with units is performed automatically.

Example

The actual fixed speed set point (CO: r1024, function diagram 3010) is to be read into the free function block ADD 0 (function diagram 7220) for further processing.

p20094[0] is set to 1024 for this purpose.

Function block ADD 0 is to be called all 128ms and is, therefore, assigned to runtime group 5.

p20096 is set to 5.

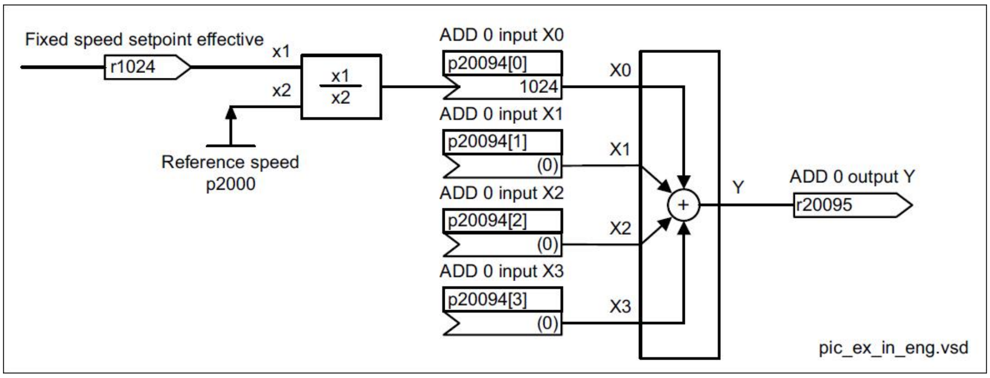

Figure 1-1: Interconnecting the speed reference value

Input signal r1024 with the unit rpm is referred to its reference variable p2000.

Assumption:

- r1024 = 1500 rpm
- p2000 = 3000 rpm reference speed

Result:

- r20095 = 0.5

1.6 Application Example

A conveyor system is to be configured in such a way that it can only start when two signals are present simultaneously. These could be the following signals, for example:

- The oil pump is running (the required pressure level is not reached, however, until after 5 seconds)
- The protective door is closed

To implement this task, you must insert free function blocks between digital input 0 and the command to switch on the motor (ON/OFF1).

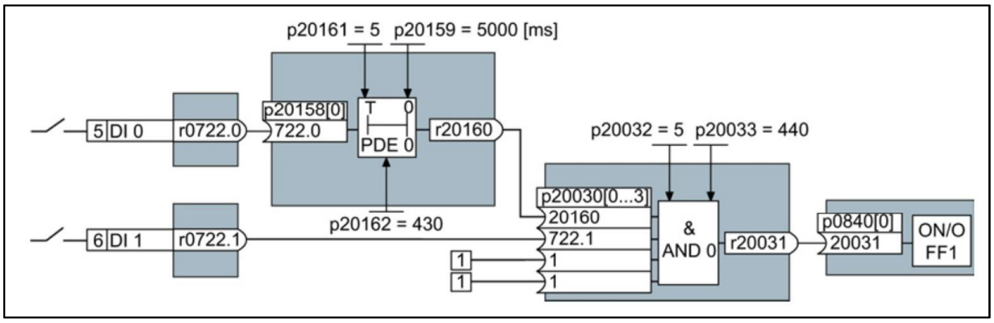

Figure 1-2: Application example of the free function blocks

The signal of digital input 0 (DI 0) is fed through a time block (PDE 0) and is interconnected with the input of a logic block (AND 0). The signal of digital input 1 (DI 1) is interconnected to the second input of the logic block. The logic block output issues the ON/OFF1 command to switch-on the motor.

Setting the control logic

Parameter	Functional description
p20161 = 5	The time block is enabled by assigning to runtime group 5 (time slice of 128 ms)
p20162 = 430	Run sequence of the time block within runtime group 5 (processing before the AND logic block)
p20032 = 5	The AND logic block is enabled by assigning to runtime group 5 (time slice of 128 ms)
p20033 = 440	Run sequence of the AND logic block within runtime group 5 (processing after the time block)
p20159 = 5000.00	Setting the delay time [ms] of the time module: 5 seconds
p20158 = 722.0	Connect the status of DI 0 to the input of the time block r0722.0 = Parameter that displays the status of digital input 0
p20030[0] = 20160	Interconnecting the time block to the 1st input of the AND
p20030[1] = 722.1	Interconnecting the status of DI 1 to the 2nd AND input r0722.1 = Parameter that displays the status of digital input 1
p0840 = 20031	Interconnect the AND output to ON/OFF1

Table 3: Control logic of the application example

2 Function Diagrams

2.1 Logic Function Blocks

2.1.1 AND - Function Block with 4 Inputs

Function diagram

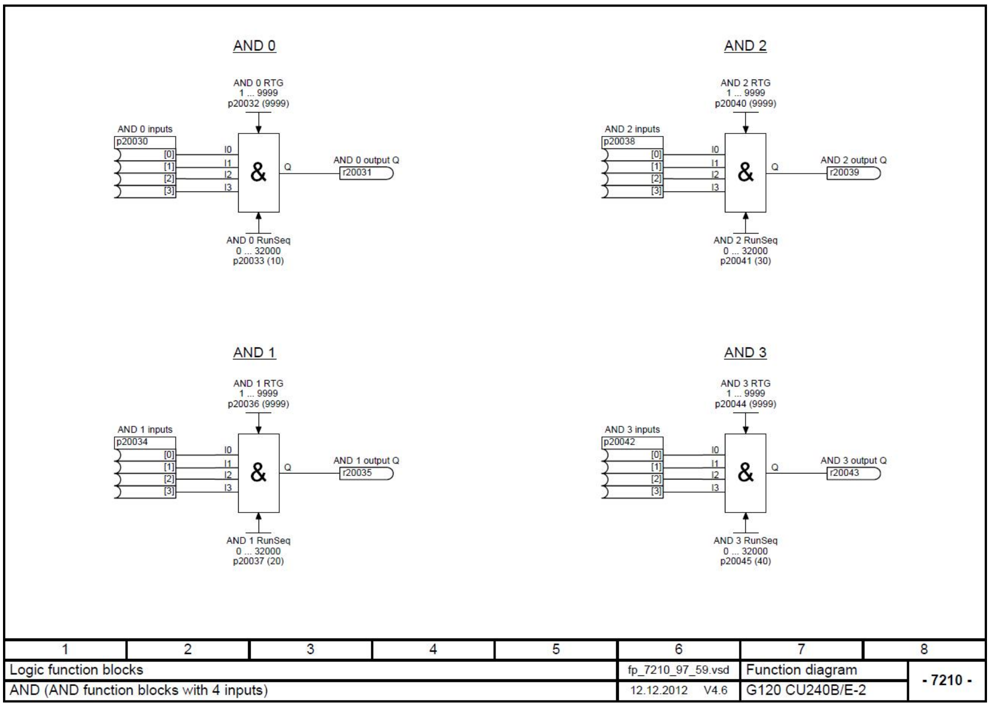

Figure 2-1: Function diagram AND operation

Functional description

This function block logically ANDs the binary variables at inputs I and outputs the result to its digital output Q.

$$Q = I0 \wedge I1 \wedge I2 \wedge I3$$

Output Q = 1 when the value 1 is present at all inputs I0 to I3. In all other cases, output Q = 0.

2.1.2 OR – Function Block with 4 Inputs

Function diagram

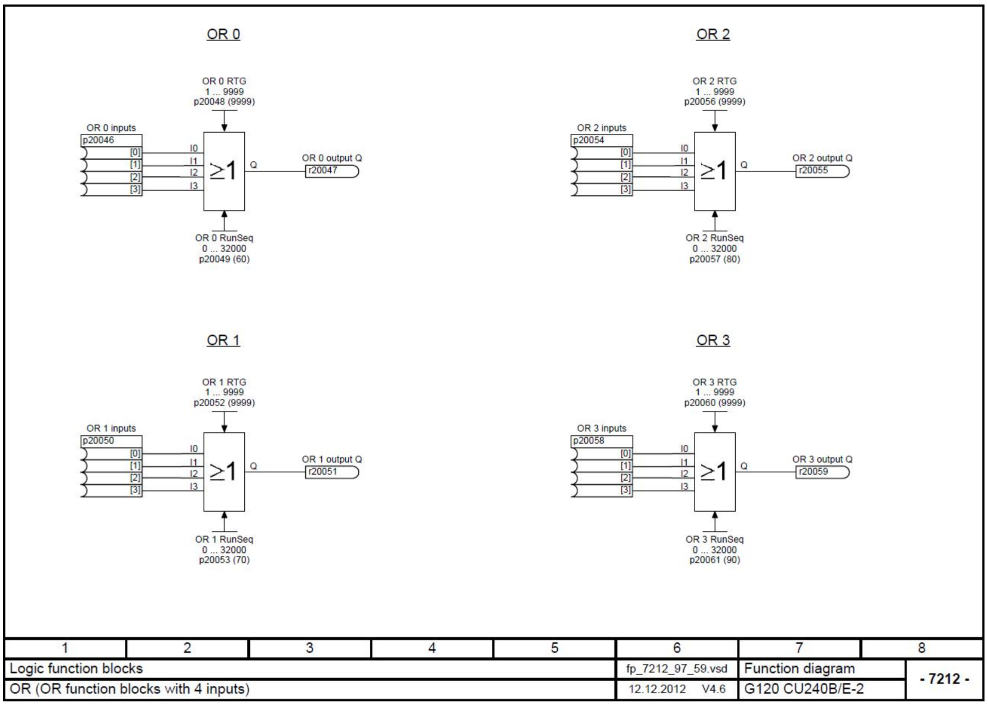

Figure 2-2: Function diagram OR operation

Functional description

This function block logically ORs the binary variables at inputs I (disjunction) and outputs the result to its digital output Q.

$$Q = I0 \vee I1 \vee I2 \vee I3$$

Output Q = 0 when the value 0 is present at all inputs I0 to I3. In all other cases, output Q = 1.

2.1.3 XOR – Function Block with 4 Inputs

Function diagram

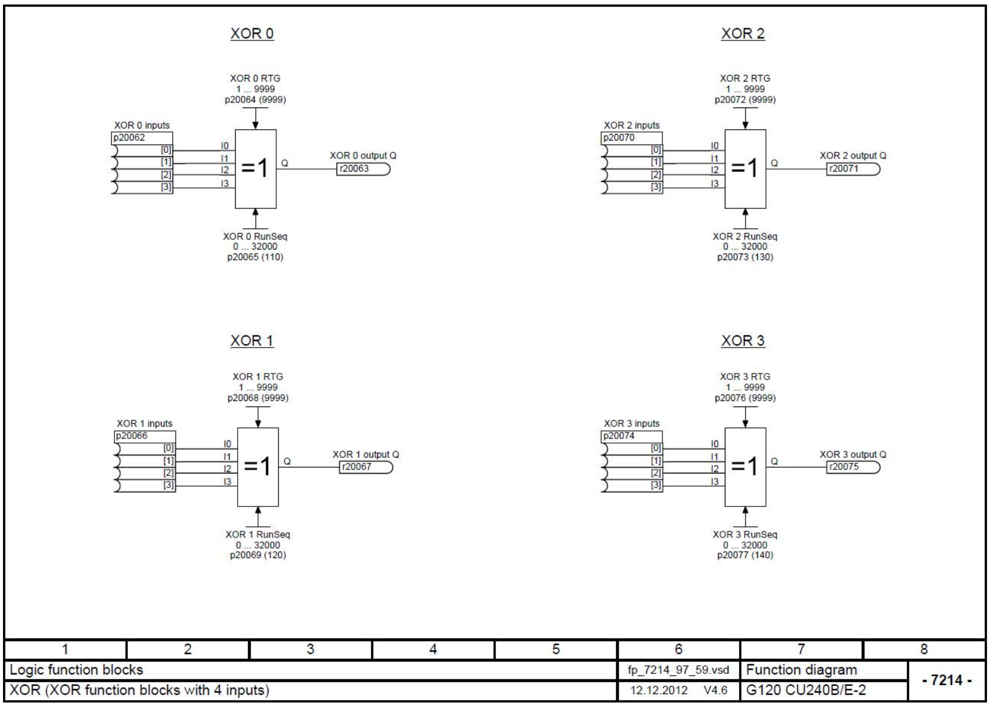

Figure 2-3: Function diagram XOR operation

Functional description

This function block links the binary variables at the inputs I according to the exclusive OR logic function and outputs the result to its digital output Q.

$$Q = I0 \veebar I1 \veebar I2 \veebar I3$$

Output Q = 0 when the value 0 is present at every input from I0 to I3 or when the value 1 is present at an even number of inputs from I0 to I3.

Output Q = 1 when the value 1 is present at an odd number of inputs from I0 to I3.

Truth table

I0	I1	I2	I3	Q
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Table 4: Truth table XOR operation

2.1.4 NOT – Inverter

Function diagram

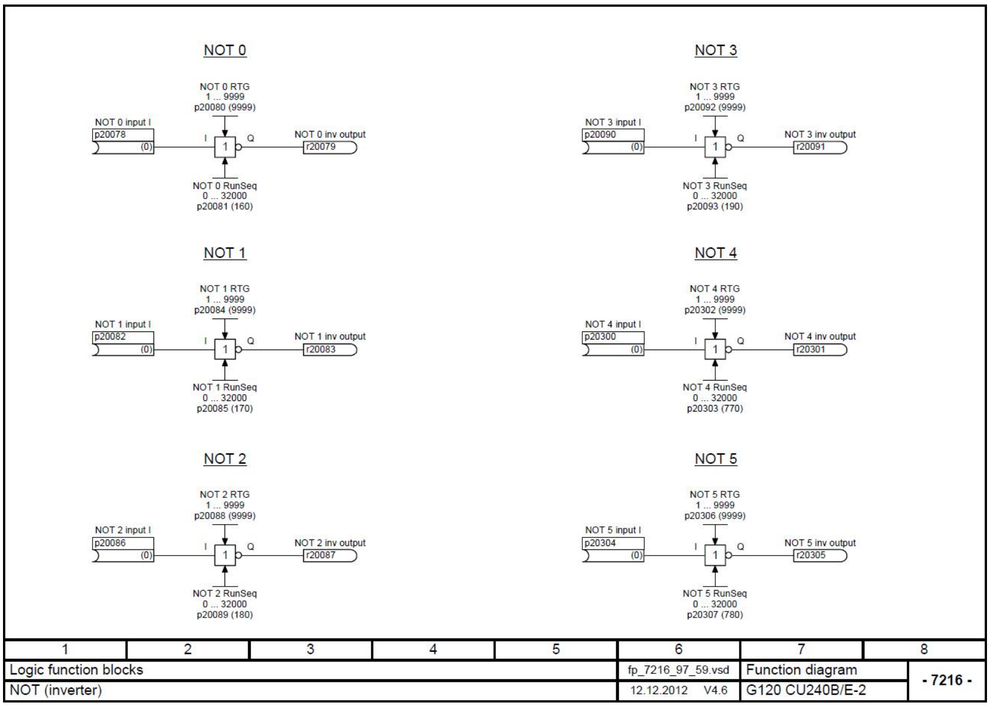

Figure 2-4: Function diagram inverter

Functional description

This function block inverts the binary variables at input I and outputs the result to output Q.

$$Q = \bar{I}$$

Output Q = 1 when the value 0 is present at input I.

Output Q = 0 when the value 1 is present at input I.

2.2 Arithmetic Function Blocks

2.2.1 ADD – Adder with 4 Inputs

Function diagram

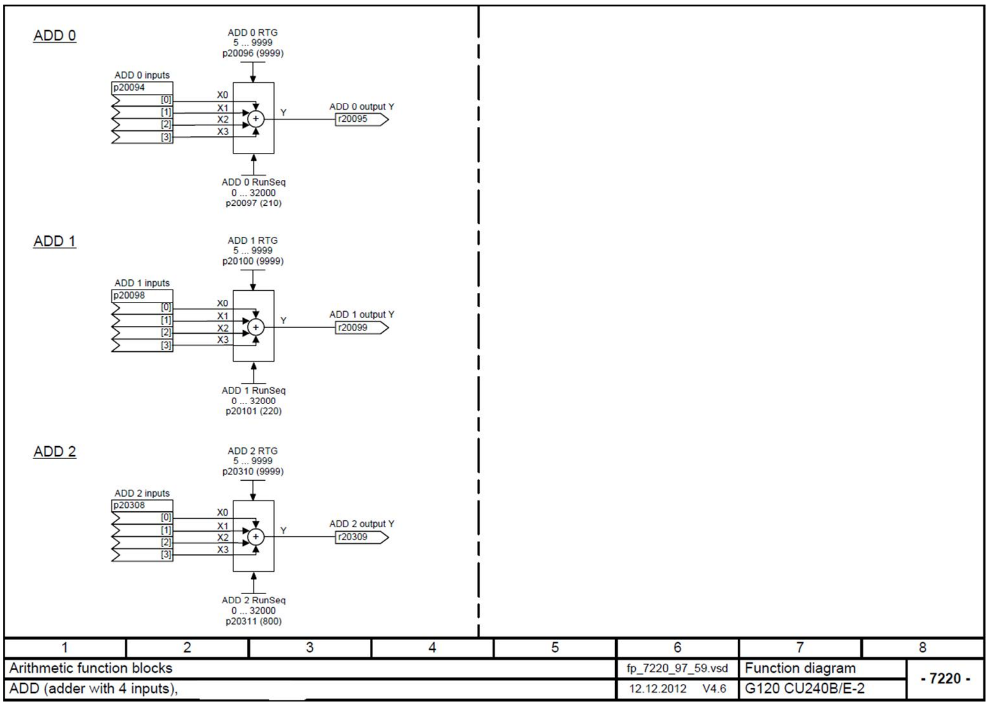

Figure 2-5: Function diagram adder

Functional description

This function block adds the values entered at inputs X, taking into account the sign.

The result is limited to a range of -3.4E38 ... 3.4E38 and output at output Y.

$$Y = X0 + X1 + X2 + X3$$

2.2.2 SUB – Subtractor

Function diagram

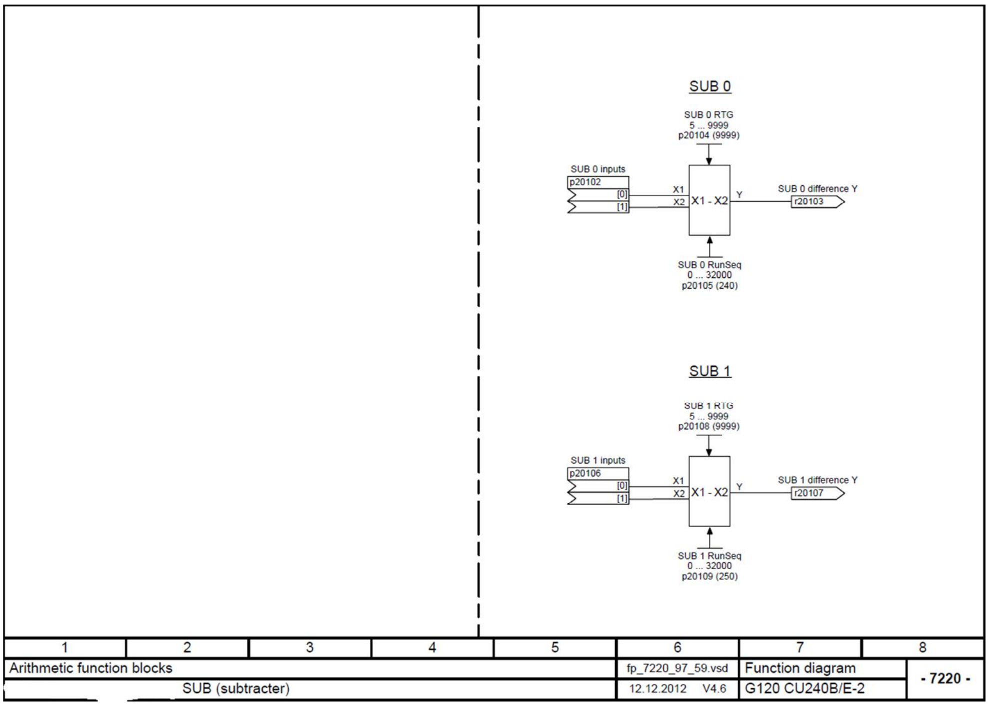

Figure 2-6: Function diagram subtractor

Functional description

This function block subtracts (taking into account the sign) the value entered at input X1 from the value entered at input X0.

The result is limited to a range of -3.4E38 ... 3.4E38 and output at output Y.

$$Y = X0 - X1$$

2.2.3 MUL – Multiplier

Function diagram

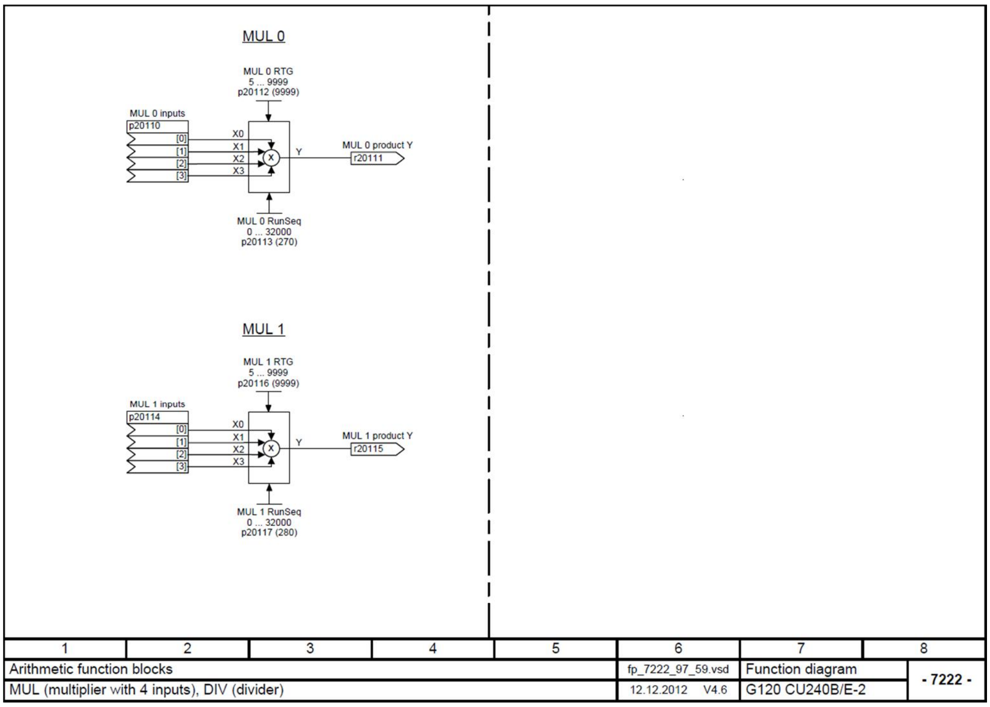

Figure 2-7: Function diagram multiplier

Functional description

This function block multiplies (taking into account the sign) the values entered at inputs X.

The result is limited to a range of -3.4E38 ... +3.4E38 and output at output Y.

$$Y = X0 * X1 * X2 * X3$$

2.2.4 DIV – Divider

Function diagram

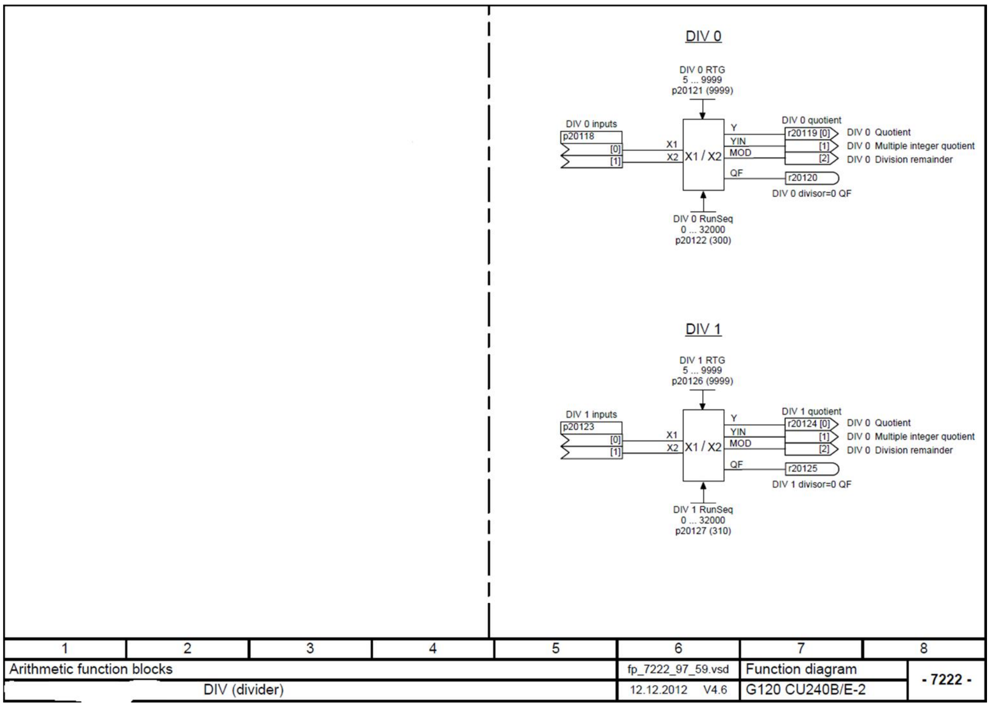

Figure 2-8: Function diagram divider

Functional description

This function block divides the value entered at input X0 by the value entered at input X1. The result is output at the outputs as follows:

- Y output: Quotient with places before and after the decimal point
- YIN output: Integer quotient
- MOD output: Division remainder (absolute remainder value, MOD = (Y - YIN) x X0)

The Y output is limited to a range of -3.4E38 ... +3.4E38.

$$Y = \frac{X1}{X2}$$

If the output value Y exceeds the permissible value range of -3.4E38 ... 3.4E38 (because divisor X1 is very small or zero), the limit value of the output range with the correct sign is output at the Y output. At the same time, digital output QF is set to 1.

With a division of 0/0, block output Y remains unchanged. Digital output QF is set to 1.

2.2.5 AVA – Absolute Value Generator

Function diagram

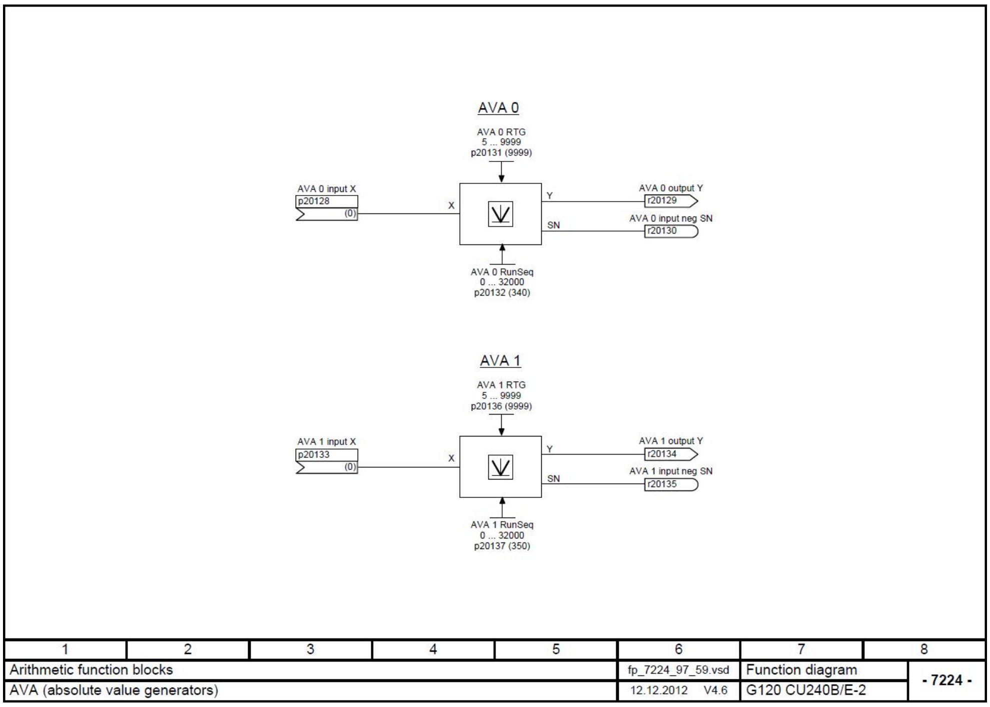

Figure 2-9: Function diagram absolute value generator

Functional description

This function block generates the absolute value of the value present at input X. The result is output at output Y.

$$Y = |X|$$

If the input variable is negative, digital output SN is set to 1.

2.2.6 NCM – Numerical Comparator

Function diagram

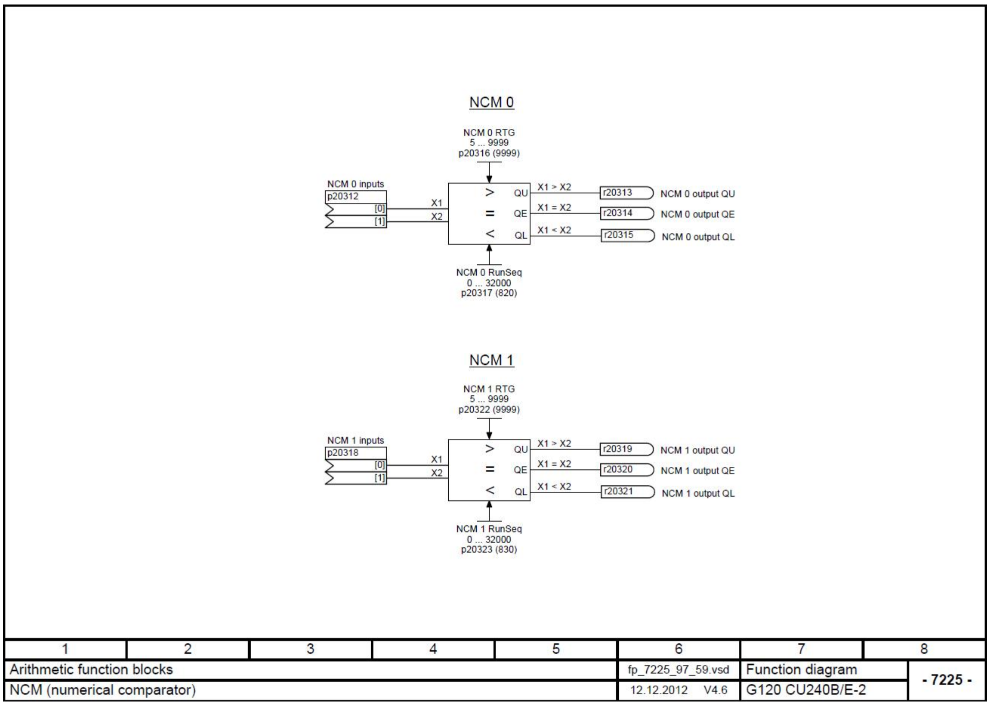

Figure 2-10: Function diagram numerical comparator

Functional description

Block for compare operations of two numeric variables. The input variables X1 and X2 are compared and one of the binary outputs QU, QE, or QL is set depending on the result of the compare operation.

Truth table

Comparison of the input variables	Output signals		
	QU	QE	QL
X1 > X2	1	0	0
X1 = X2	0	1	0
X1 < X2	0	0	1

Table 5: Truth table numerical comparator

2.2.7 PLI – Polyline

Function diagram

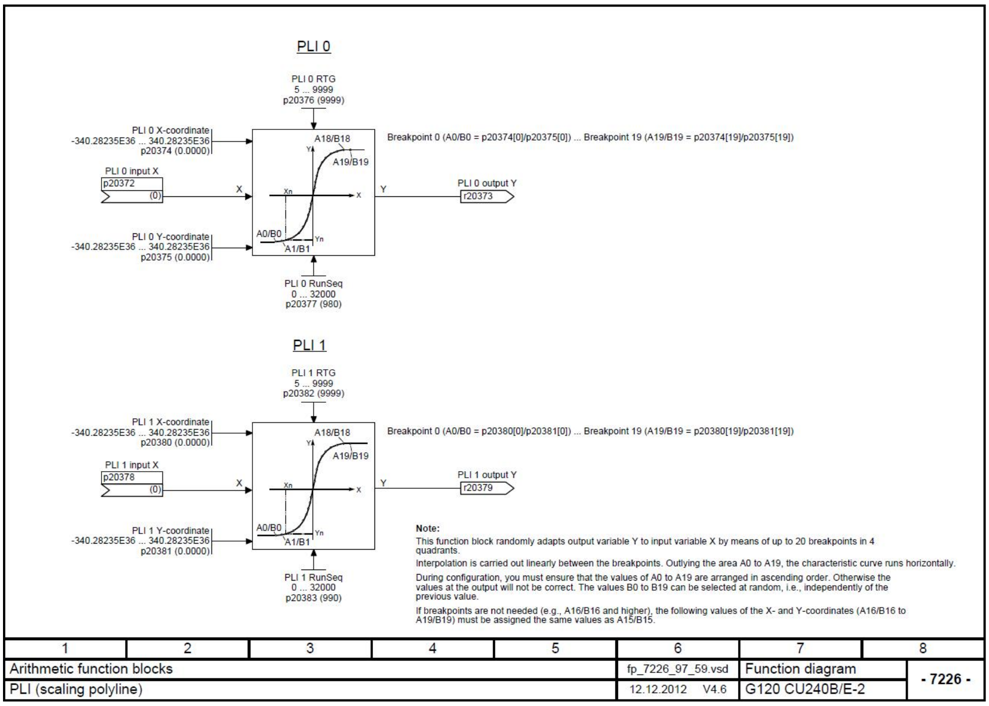

Figure 2-11: Function diagram polyline

Functional description

Function block for linearization of characteristic curves, as

- for simulation of non-linear transfer elements
- for controller gain defined in sections

This block arbitrarily adapts the output variable Y to the input variable X via max. 20 breakpoints in 4 quadrants. Interpolation is carried out linearly between the breakpoints. The characteristic is horizontal outside of A1 or A20..

Configuration guide

During configuration, you must ensure that the values of A1 to A20 are sorted in ascending order otherwise incorrect values are output. The ordinate values B1 to B20 can be selected arbitrarily, i.e. irrespective of the preceding value.

If breakpoints are not needed (e.g. as of A16/B16), the following abscissas and ordinates (A16/B16 to A20/B20) must be assigned the same values as A15/B15.

Example

The analog output AO0 shall output a voltage signal in dependency of the drive speed according the following scheme:

- speed 0 to 30%, output voltage 2V
- speed 30 to 60%, output voltage 3V
- speed 60 to 100%, output voltage 5V

The interconnection of the function block is as following:

Parameter	Value
p20372	r0021, n_act smooth[rpm]
p0771[0]	r20373, PLI 0 output Y
p20376	Runtime group 5, 128ms
p20377	Run sequence 980

Table 6: Connection PLI0 function block

The breaking points can be assigned in two different variants. The first variant shows a linear distribution of from 0 to 100%, variant 2 a non-linear distribution. After interpolation between the breaking points, the latter has the advantage that the output voltage is switched quickly to the new value between the breaking points and no unwanted intermediate values occur.

Parameter	Value	Parameter	Value
p20374[0]	0,0500	p20375[0]	0,2000
p20374[1]	0,1000	p20375[1]	0,2000
p20374[2]	0,1500	p20375[2]	0,2000
p20374[3]	0,2000	p20375[3]	0,2000
p20374[4]	0,2500	p20375[4]	0,2000
p20374[5]	0,3000	p20375[5]	0,2000
p20374[6]	0,3500	p20375[6]	0,3000
p20374[7]	0,4000	p20375[7]	0,3000
p20374[8]	0,4500	p20375[8]	0,3000
p20374[9]	0,5000	p20375[9]	0,3000
p20374[10]	0,5500	p20375[10]	0,3000
p20374[11]	0,6000	p20375[11]	0,3000
p20374[12]	0,6500	p20375[12]	0,5000
p20374[13]	0,7000	p20375[13]	0,5000
p20374[14]	0,7500	p20375[14]	0,5000
p20374[15]	0,8000	p20375[15]	0,5000
p20374[16]	0,8500	p20375[16]	0,5000
p20374[17]	0,9000	p20375[17]	0,5000
p20374[18]	0,9500	p20375[18]	0,5000
p20374[19]	1,0000	p20375[19]	0,5000

Table 7: Parameterization of the breaking points at linear distribution

Parameter	Value	Parameter	Value
p20374[0]	0,0000	p20375[0]	0,2000
p20374[1]	0,3000	p20375[1]	0,2000
p20374[2]	0,3001	p20375[2]	0,3000
p20374[3]	0,6000	p20375[3]	0,3000
p20374[4]	0,6001	p20375[4]	0,5000
p20374[5]	1,0000	p20375[5]	0,5000
p20374[6]	1,0000	p20375[6]	0,5000
p20374[7]	1,0000	p20375[7]	0,5000
p20374[8]	1,0000	p20375[8]	0,5000
p20374[9]	1,0000	p20375[9]	0,5000
p20374[10]	1,0000	p20375[10]	0,5000
p20374[11]	1,0000	p20375[11]	0,5000
p20374[12]	1,0000	p20375[12]	0,5000
p20374[13]	1,0000	p20375[13]	0,5000
p20374[14]	1,0000	p20375[14]	0,5000
p20374[15]	1,0000	p20375[15]	0,5000
p20374[16]	1,0000	p20375[16]	0,5000
p20374[17]	1,0000	p20375[17]	0,5000
p20374[18]	1,0000	p20375[18]	0,5000
p20374[19]	1,0000	p20375[19]	0,5000

Table 8: Parameterization of the breaking points at nonlinear distribution

If only a part of the breaking points A0 to A19 are necessary, it has to be made sure that the values above the last " necessary " breaking point (breaking point A5 in the example) must be assigned the values of the last breaking point , to prevent malfunctions.

2.3 Time Function Blocks

2.3.1 MFP – Pulse Generator

Function diagram

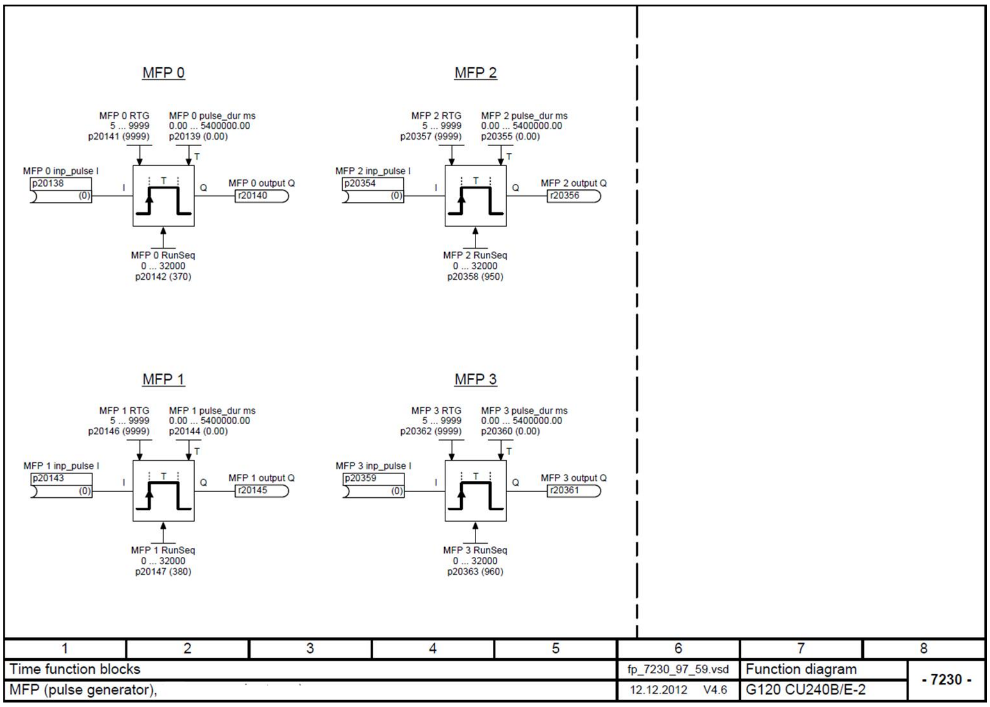

Figure 2-12: Function diagram pulse generator

Functional description

The pulse generator is a timer for generating a pulse with a fixed duration. It can be used as a pulse-contracting or pulse-stretching monoflop..

The rising edge of a pulse at the input I sets the output Q to 1 for the pulse duration T. The pulse generator cannot be retriggered.

Time sequence chart

Output pulse Q as a function of pulse duration T and input pulse I.

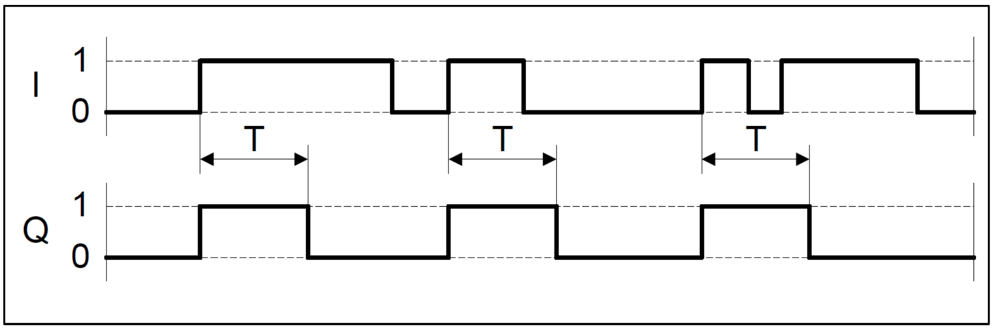

Figure 2-13: Time sequence pulse generator

2.3.2 PCL – Pulse Contractor

Function diagram

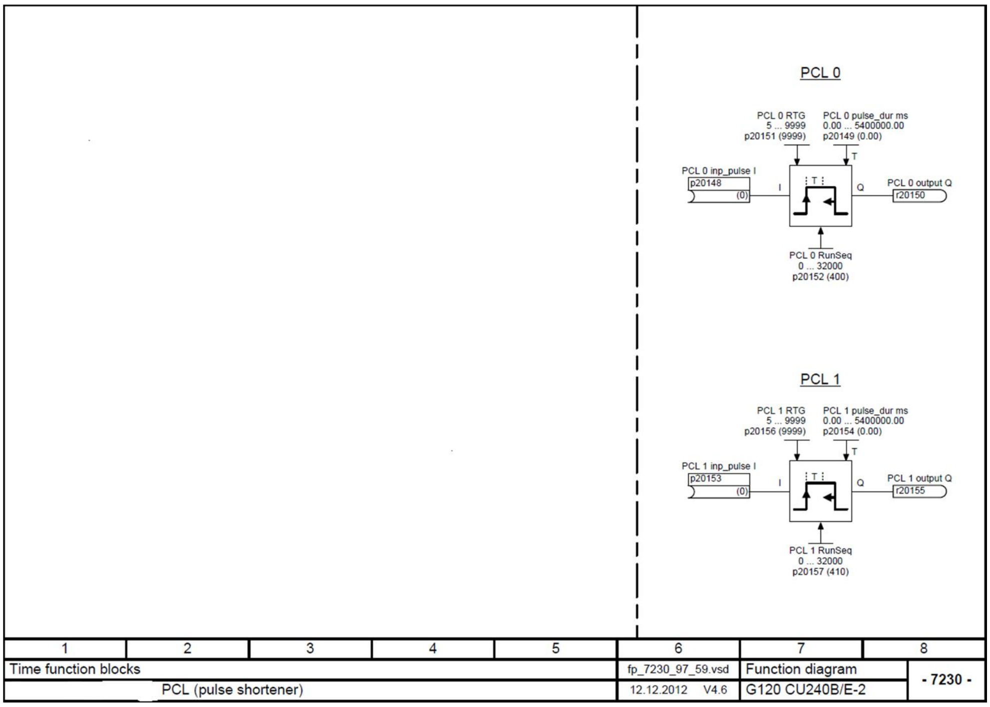

Figure 2-14: Function diagram pulse contractor

Functional description

The pulse contractor is a timer for limiting the pulse duration. The rising edge of a pulse at input I sets the output Q to 1.

Output Q becomes 0 when input I is 0 or pulse duration T has expired..

Time sequence chart

Output pulse Q as a function of pulse duration T and input pulse I.

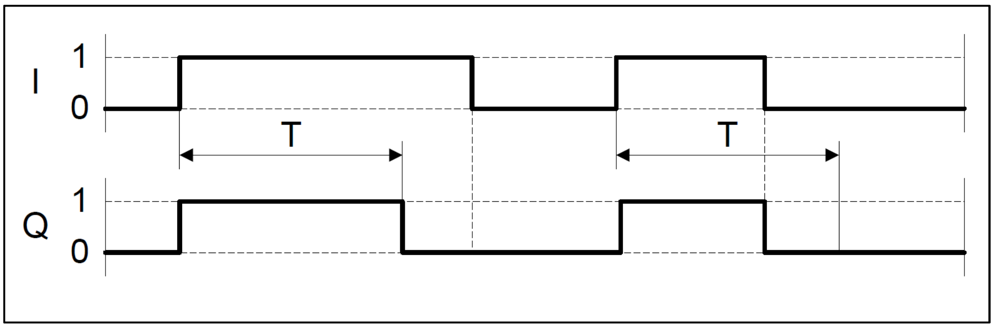

Figure 2-15: Time sequence pulse contractor

2.3.3 PDE – Switch ON Delay

Function diagram

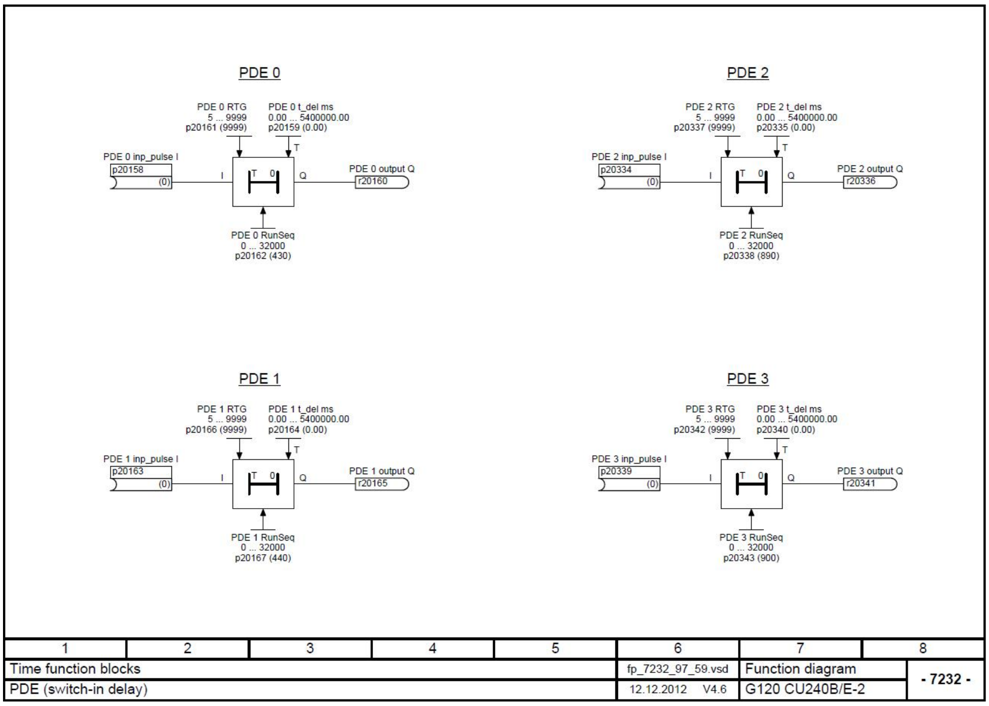

Figure 2-16: Function diagram switch ON delay

Functional description

Timer with parameterizable switch ON delay. The rising edge of a pulse at input I sets the output Q to 1 after the pulse delay time T.

Output Q becomes 0 when I is 0.

If the duration of the input pulse I is less than the pulse delay time T, Q remains 0.

Time sequence chart

Output pulse Q as a function of pulse duration T and input pulse I.

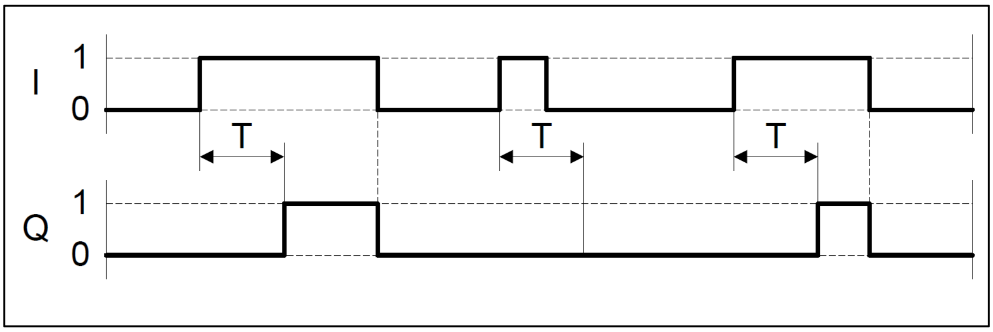

Figure 2-17: Time sequence switch ON delay

2.3.4 PDF – Switch OFF Delay

Function diagram

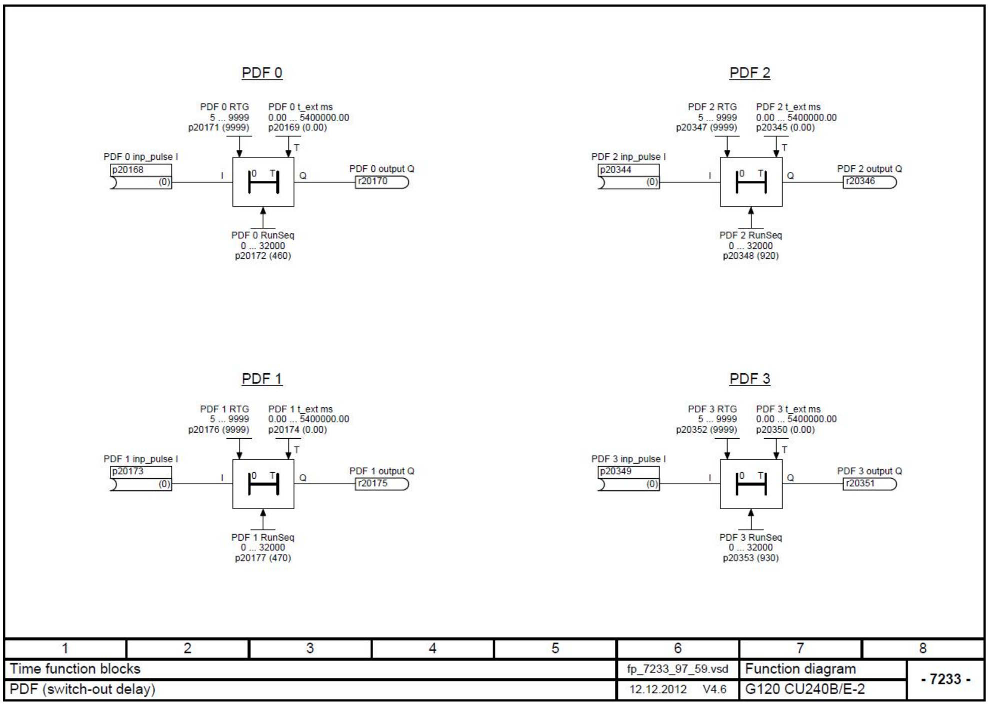

Figure 2-18: Function diagram switch OFF delay

Functional description

Timer with parameterizable switch OFF delay. The falling edge of a pulse at input I resets the output Q to 0 after the OFF delay time T.

Output Q becomes 1 when I is 1.

Output Q becomes 0 when input pulse I is 0 and the OFF delay time T has expired.

If input I is reset to 1 before time T has expired, output Q remains 1.

Time sequence chart

Output pulse Q as a function of pulse duration T and input pulse I.

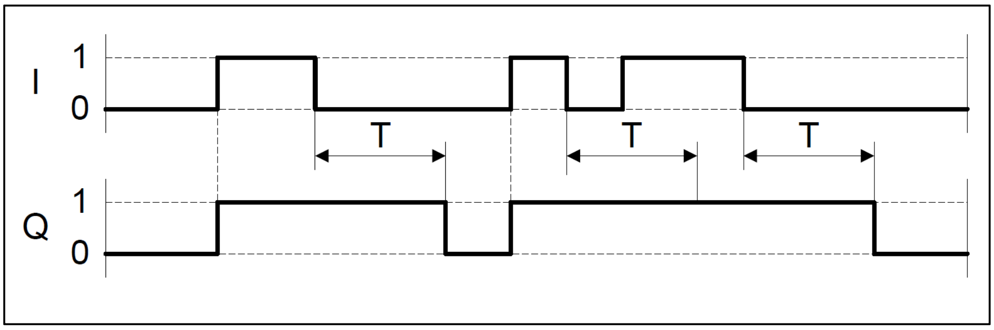

Figure 2-19: Time sequence switch OFF delay

2.3.5 PST – Pulse Stretcher

Function diagram

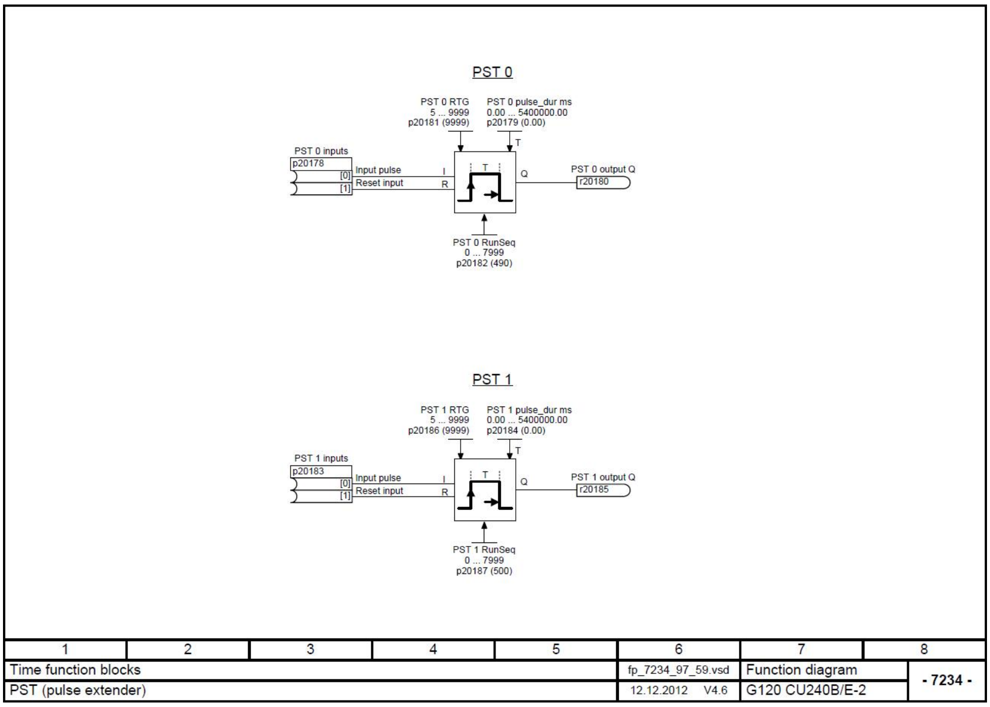

Figure 2-20: Function diagram pulse stretcher

Functional description

Timer for generating a pulse with a minimum duration and an additional reset input. The rising edge of a pulse at input I sets output Q to 1.

Output Q does not return to 1 until input pulse I is 0 and pulse duration T has expired.

Output Q can be set to zero at any time via reset input R with R = 1

Time sequence chart

Output pulse Q as a function of pulse duration T and input pulse I (when R = 0).

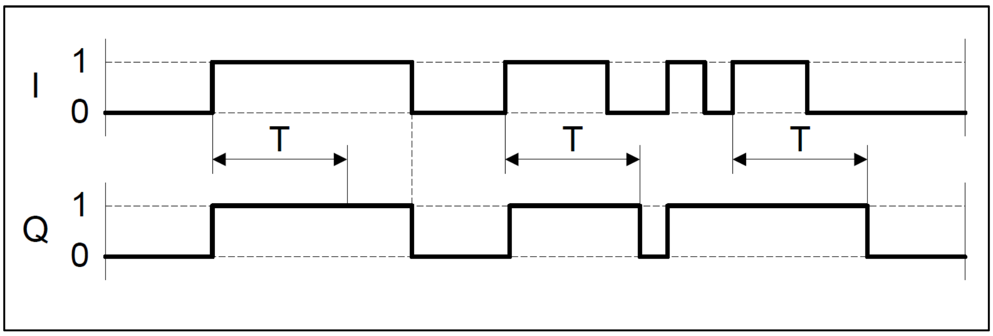

Figure 2-21: Time sequence pulse stretcher

2.4 Memory Function Blocks

2.4.1 RSR – RS Flip-Flop

Function diagram

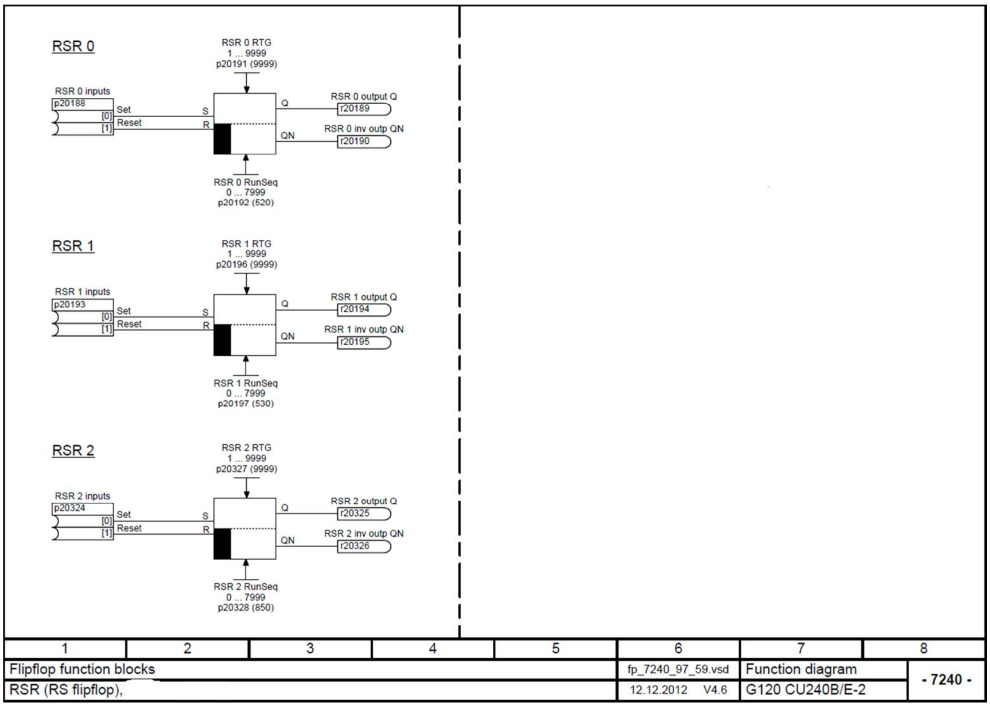

Figure 2-22: Function diagram RS flip-flop

Functional description

Reset dominant RS flip-flop for use as a static binary value memory. With logical 1 at input S, output Q is set to logical 1. If input R is set to logical 1, output Q is set to logical 0. If both inputs are logical 0, Q does not change. If both inputs are logical 1, however, Q is logical 0 because the reset input dominates.

Output QN always has the opposite value to Q.

Truth table

Inputs		Outputs	
Set	Reset	Q	QN
0	0	Q_n	$\overline{Q_n}$
1	0	1	0
0	1	0	1
1	1	0	1

Table 9: Truth table RS flip-flop

2.4.2 DFR – D Flip-Flop

Function diagram

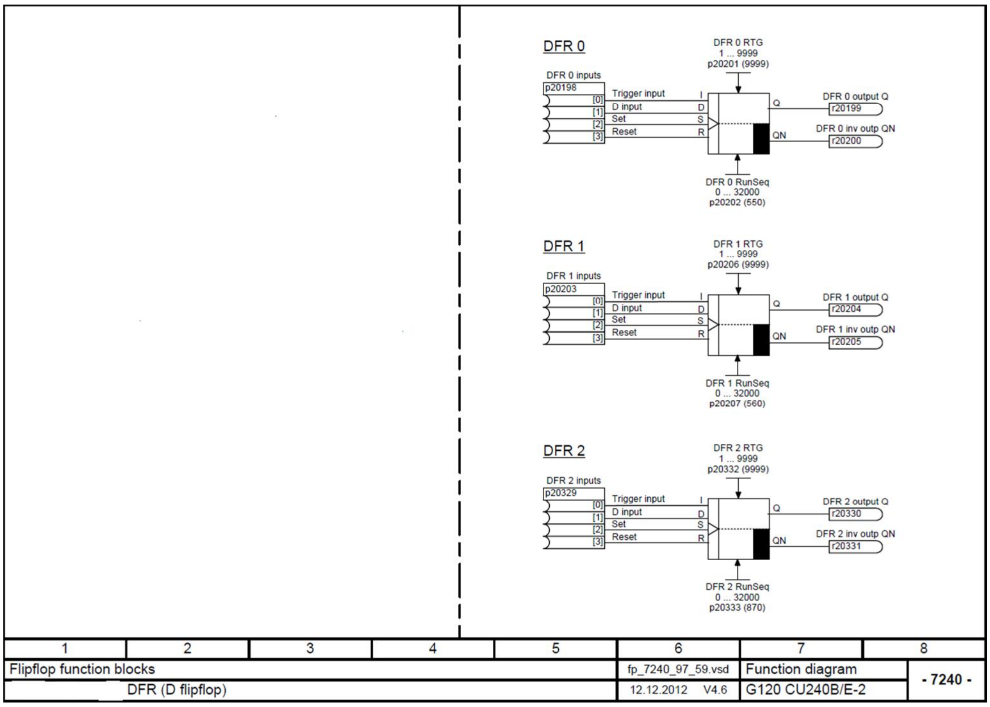

Figure 2-23: Function diagram D flip-flop

Functional description

Function block for use as a D flip-flop with reset dominance. If the inputs S and R are logical 0, the D input data is switched through to output Q, when a rising edge is present at trigger input I. Output QN always has the opposite value to Q. With logical 1 at input S, output Q is set to logical 1.

If input R is set to logical 1, output Q is set to logical 0. If both inputs are logical 0, Q does not change.

If inputs S and R are logical 1, however, Q is logical 0 because the reset input dominates.

Time sequence chart

Output pulse Q as a function of the D input and input pulse I for S = R = 0.

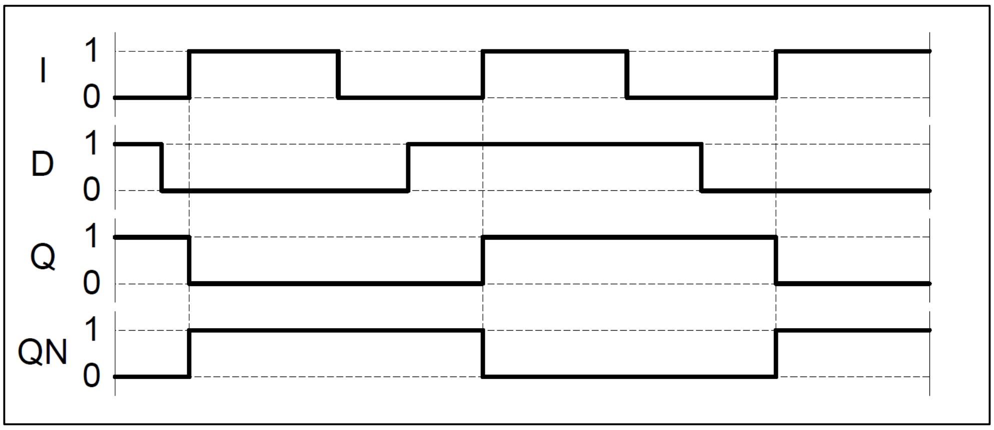

Figure 2-24: Time sequence D flip-flop

Truth table

Inputs				Outputs	
D	**Trigger**	**Set**	**Reset**	**Q**	**QN**
0	0	0	0	Q_n	$\overline{Q_n}$
1	↑	0	0	1	0
0	↑	0	0	0	1
0	0	1	0	1	0
0	0	0	1	0	1
0	0	1	1	0	1

Table 10: Truth table D flip-flop

2.5 Switch Function Blocks

2.5.1 BSW – Binary Change-Over Switch

Function diagram

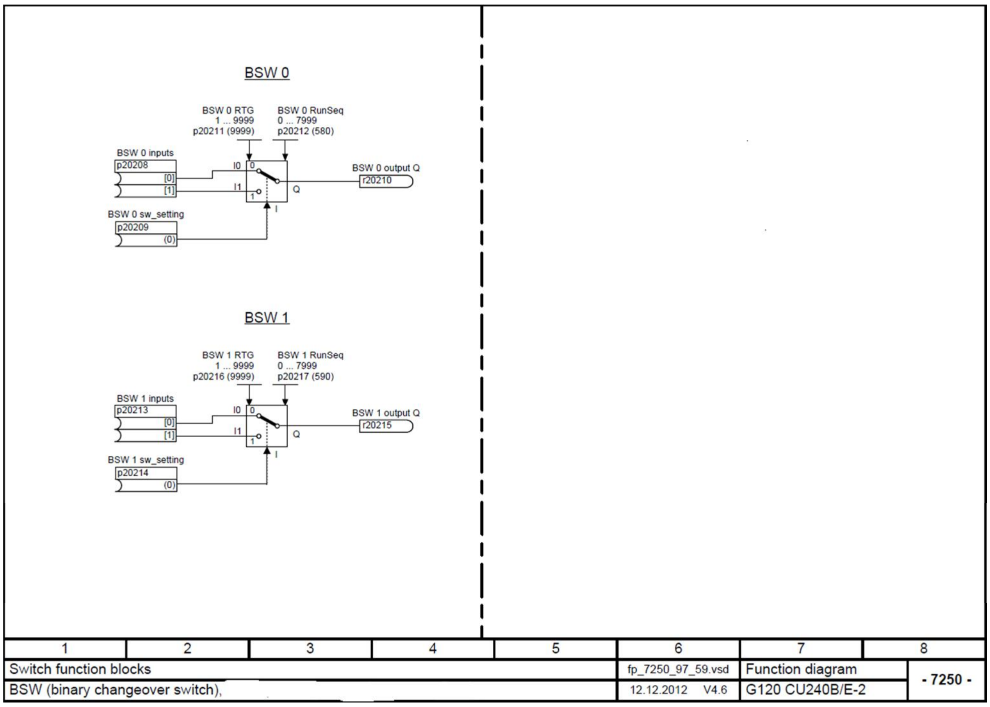

Figure 2-25: Function diagram binary change-over switch

Functional description

This function block switches one of two binary input variables to the output.

If input I = 0, I0 is switched to output Q.

If input I = 1, I1 is switched to output Q.

2.5.2 NSW – Numerical Change-Over Switch

Function diagram

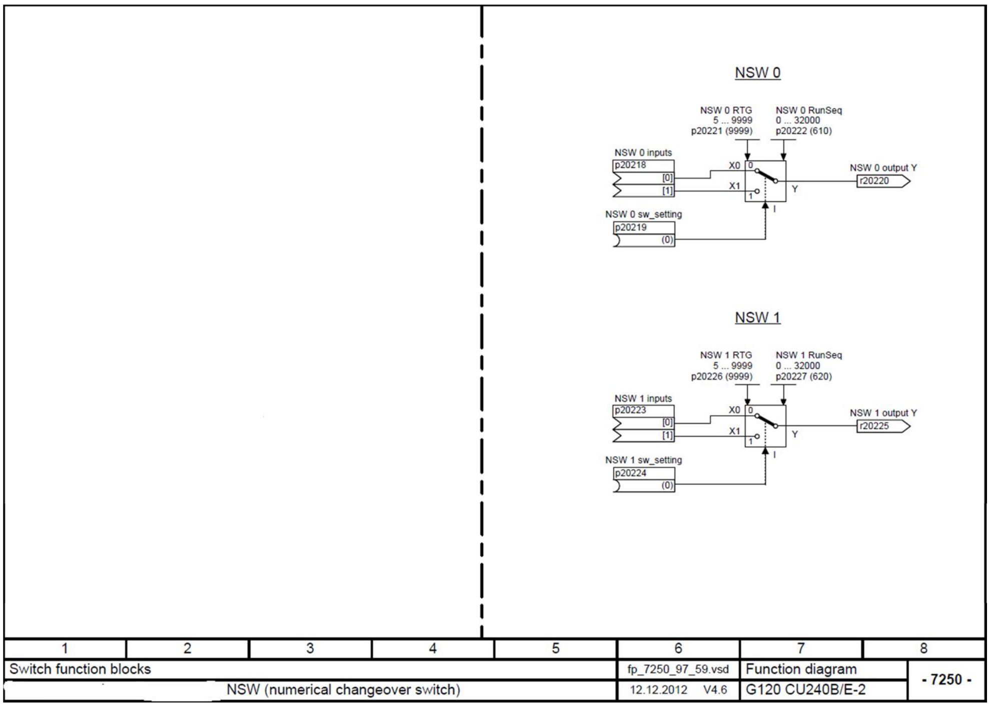

Figure 2-26: Function diagram numerical change-over switch

Functional description

This function block switches one of two numeric input variables to the output.

If input I = 0, X0 is switched to output Y.

If input I = 1, X1 is switched to output Y.

2.6 Control Function Blocks

2.6.1 LIM – Limiter

Function diagram

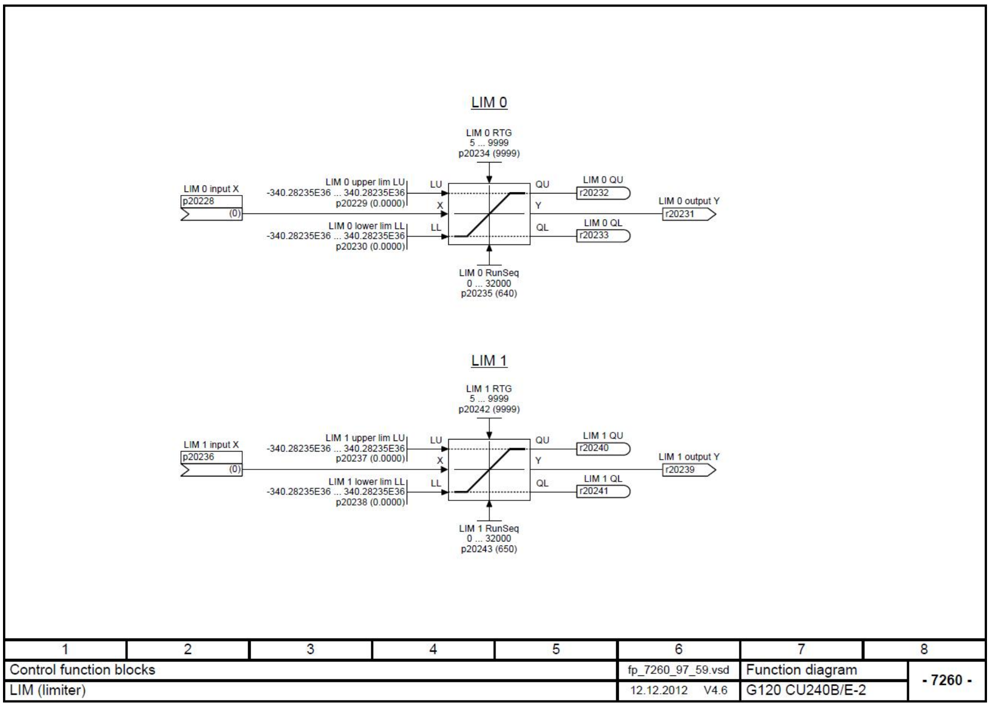

Figure 2-27: Function diagram limiter

Functional description

Function block to limit numerical input variables, with adjustable upper and lower limit as well as indication when set limits are reached.

This function block transfers the input variable X to its output Y. The input variable is limited depending on LU and LL.

If the input variable reaches the upper limit LU, output QU is set to 1.

If the input variable reaches the lower limit LL, output QL is set to 1.

If the lower limit is greater than or equal to the upper limit, output Y is set to the upper limit LU.

Algorithm:

$$Y = \begin{cases} LU \text{ für } X \geq LU \\ X \text{ für } LL < X < LU \\ LL \text{ für } X \leq LL \end{cases}$$

Supplementary condition: LL < LU

2.6.2 PT1 – Smoothing Element

Function diagram

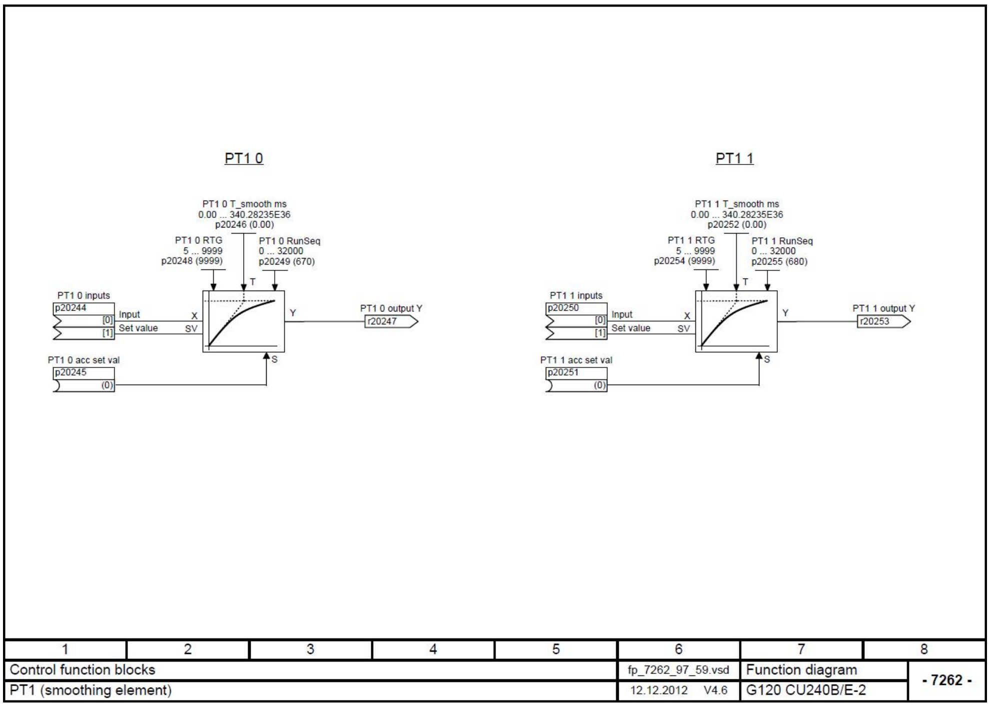

Figure 2-28: Function diagram PT1 smoothing element

Functional description

First-order delay element with setting function used as smoothing element

Setting function not active (S = 0)

Input variable X, dynamically delayed by smoothing time constant T, is switched to output Y. T determines the steepness of the rise of the output variable. It specifies the time at which the transfer function has risen to 63% of its full-scale value. When t = 3T, the transfer function reaches approximately 95% of its full-scale value. The internally fixed proportional gain is 1 and cannot be changed.

Calculation algorithm

$$Y(t) = X \cdot (1 - e^{-t/T})$$

Supplementary condition: t = n * Ts

Further functionality

Setting function active (S = 1)

- When the setting function is active, the actual setting value SVn is accepted at the output variable:
  Yn = SVn

Note If the smoothing time is chosen T ≤ Ts, the output Y changes without delay to the final value. Only with T > 2Ts it results in the typical waveform of a first order delay element.

2.6.3 INT – Integrator

Function diagram

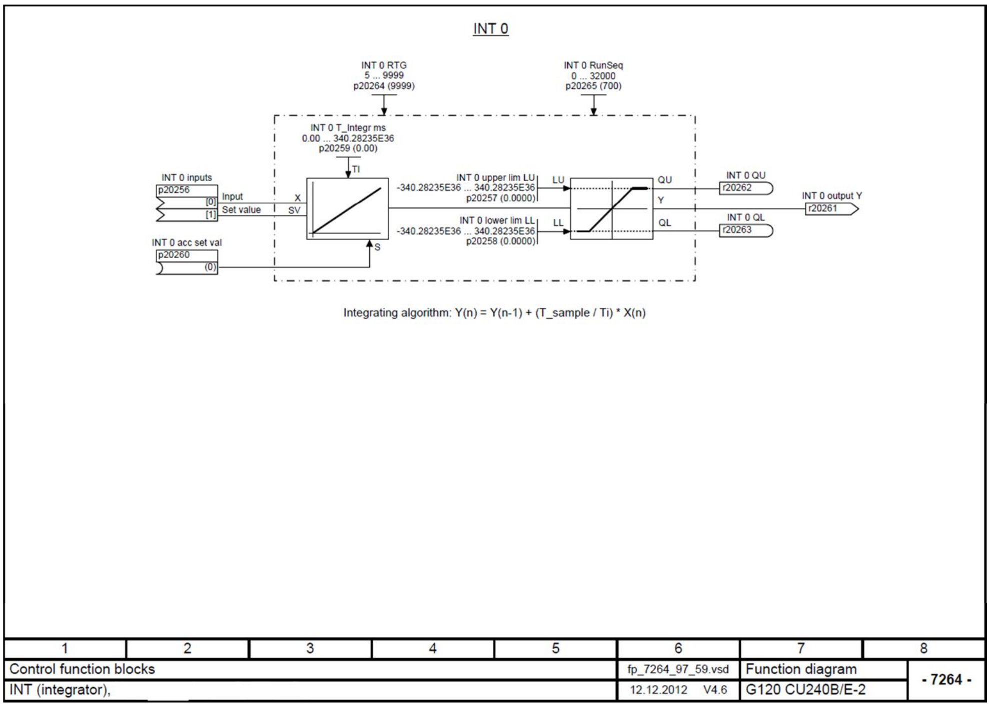

Figure 2-29: Function diagram integrator

Functional description

Function block with integrating action and the following integrator functions:

- Set initial value.
- Adjustable integral time constant.
- Adjustable limits.

The change in output variable Y is proportional to input variable X and inversely proportional to the integral time constant TI. The output Y of the integrator can be limited via inputs LU and LL. If the output reaches one of the two limits, a signal is issued via output QU or QL. If LL >= LU, output Y = LU.

Calculation algorithm

$$Y_n = Y_{n-1} + (T\_sample/T_i) * X_n$$

Further functionality

When S = 1, the output variable Y is set to the setting value SV. Two functions can be performed via S:

- Track integrator (Y = SV)
  The digital input is S = 1 and the setting value SV is changed. If applicable, the output makes a jump to the setting value immediately after the setting operation.
- Set integrator to initial value SV.
  S is switched to 1. S is then set to 0, and the integrator starts from SV in the direction specified by the polarity of input variable X.

Note If the integrating time is chosen Ti ≤ Ta, the output Y changes without delay to the final value. Only with Ti > 2Ta it results in the typical waveform of an integrator.

2.6.4 DIF – Derivative Element

Function diagram

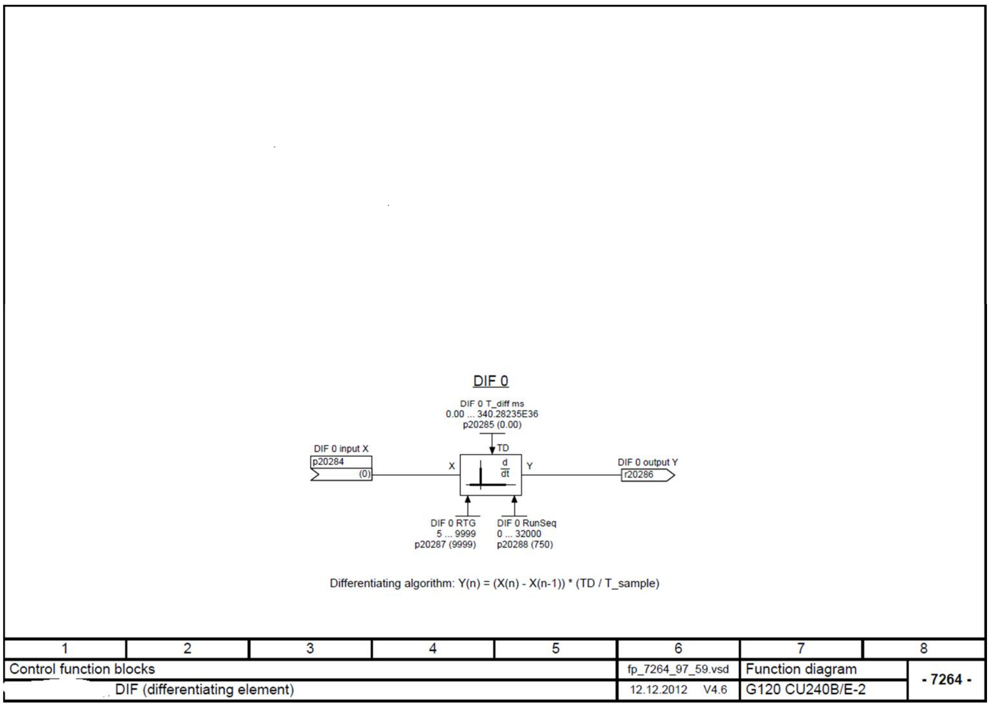

Figure 2-30: Function diagram derivative element

Functional description

Function block with derivative action behavior. Output variable Y is proportional to the rate of change of input variable X multiplied by derivative time constant TD.

Calculation algorithm

$$Y_n = (X_n - X_{n-1}) * (TD/T\_sample)$$

Note The higher that TD/Ts is, the higher the amplitude change on Y from one sampling time to the next.

Caution: Overcontrol possible!

2.7 Complex Function Blocks

2.7.1 LVM – Double-Sided Limit Monitor with Hysteresis

Function diagram

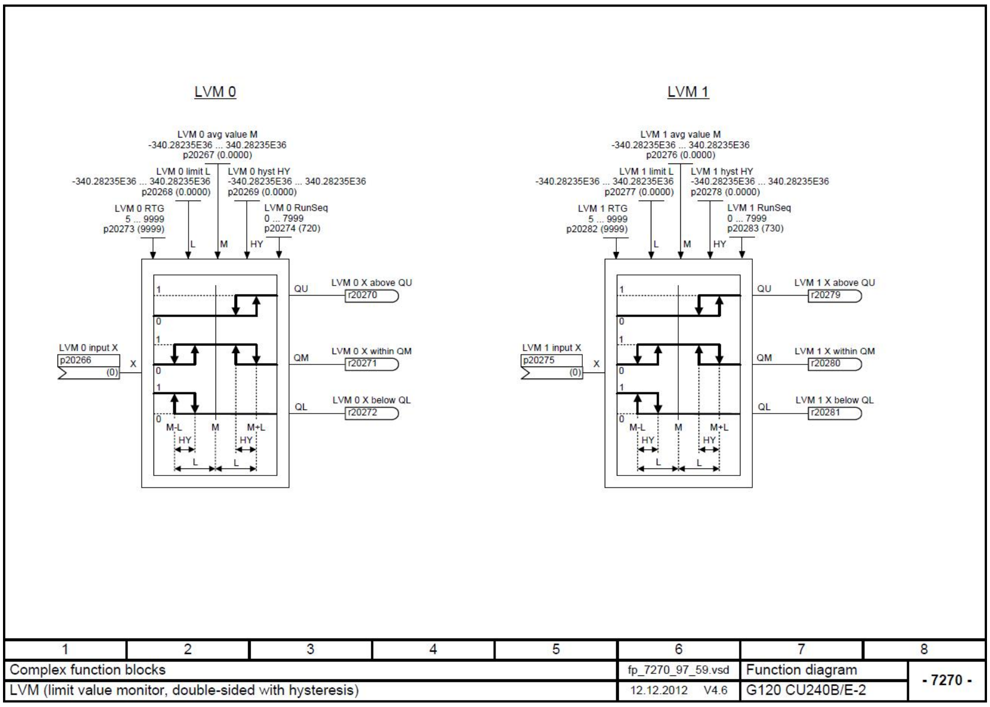

Figure 2-31:Function diagram double-sided limit monitor with hysteresis

Functional description

This function block monitors an input variable by comparing it with selectable reference variables.

Applications:

- Monitoring set points, actual, and measured values.
- Suppressing frequent switching (jitter).

This function block provides a window discriminator function. It uses a transfer characteristic (see transfer characteristic) with hysteresis to calculate an internal intermediate value. The intermediate value is compared with the interval limits and the result is output at outputs QU, QM, and QL.

The transfer characteristic is configured with the values for the mean value M, the interval limit L, and the hysteresis HY.

Transfer characteristic

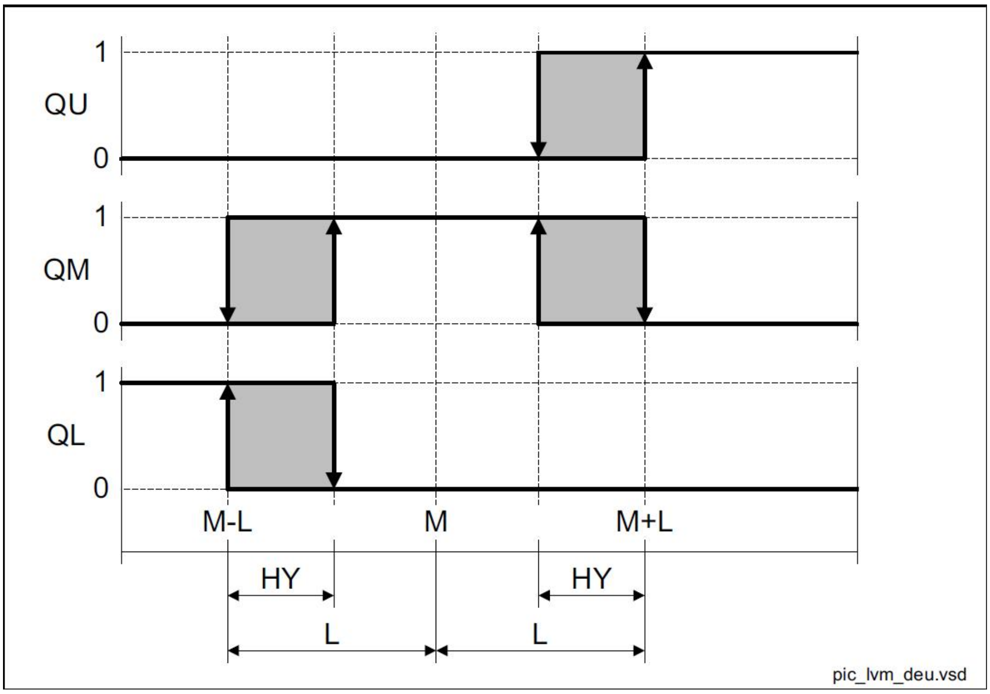

Figure 2-32: Transfer characteristic double-sided limit monitor with hysteresis

3 History

Version	Date	Change Log
V1.0	11/2013	First Edition
V1.1	02/17/16	SINAMICS G120C included

Table 11