X20DS4389

1 General information

This module is a digital signal processor module that is used for detecting and evaluating input edges and for creating edges.

In oversampling mode, the module acquires very short input patters whose low or high phases are shorter than the X2X Link cycle time. Similarly, output patterns (e.g. drum sequencers) can also be output with extremely short high/low times. Oversampling can take place with a scan rate of up to 25 μs.

If necessary, up to 4 events per edge detection unit can be stored in a buffer (history elements).

Other functions include pulse duration measuring and differential time measuring.

- 4 digital input channels
- 4 digital channels, configurable as inputs or outputs
- 4 edge detection units with timestamp function (each usable as pulse duration or differential time measurement, 4 history elements per unit)
- 4x precise edge generation down to the μs (up to four edges per unit in each case)
- 4x oversampling (input and output signal)
- 24 VDC and GND for sensor/actuator supply
- NetTime timestamp: Input data, edge detection, edge generation

NetTime-Zeitstempel

An additional essential feature is the module's integrated timestamp function. This allows fast input edges such as registration marks to be detected independently of the system's X2X Link cycle time and provided with a precise input stamp. In the other direction, the module sets outputs at exactly specified times. This is done with a resolution up to 125 ns.

1.1 Other applicable documents

For additional and supplementary information, see the following documents.

Other applicable documents

Document name	Title
MAX20	X20 system user's manual
MAEMV	Installation / EMC guide

2 Order data

Order number	Short description	Figure
	Digital signal processing and preparation	
X20DS4389	X20 digital signal module, 4 digital inputs, 24 VDC, 4 digital outputs, 24 VDC, 0.1 A, oversampling I/O functions, time-triggered I/O functions, NetTime function	
	Required accessories	
	Bus modules	
X20BM11	X20 bus module, 24 VDC keyed, internal I/O power supply connected through	
X20BM15	X20 bus module, with node number switch, 24 VDC keyed, internal I/O power supply connected through	
	Terminal blocks	
X20TB12	X20 terminal block, 12-pin, 24 VDC keyed	

Table 1: X20DS4389 - Order data

3 Technical description

3.1 Technical data

Order number	X20DS4389
Short description	
I/O module	4 digital input channels, 4 digital channels configurable as inputs or outputs, 4 edge detection units with timestamp function (each usable as pulse duration or differential time measurement, 4 history elements per unit), 4x microsecond-accurate edge generation (each up to 4 edges per unit), 4x oversampling (input and output signal)
General information	
B&R ID code	0xA93B
Status indicators	I/O function per channel, operating state, module status
Diagnostics	
Module run/error	Yes, using LED status indicator and software
Outputs	Yes, using LED status indicator and software (output state)
Power consumption	
Bus	0.01 W
Internal I/O	1.5 W
Additional power dissipation caused by actuators (resistive) [W]	-
Type of signal lines	Shielded lines must be used for all signal lines.
Certifications	
CE	Yes
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X
UL	cULus E115267 Industrial control equipment
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5
DNV	Temperature: **B** (0 - 55°C) Humidity: **B** (up to 100%) Vibration: **B** (4 g) EMC: **B** (bridge and open deck)
LR	ENV1
KR	Yes
ABS	Yes
EAC	Yes
KC	Yes
Digital inputs	
Quantity	4 + 4, configuration as input or output using software
Nominal voltage	24 VDC
Input voltage	24 VDC -15% / +20%
Input current at 24 VDC	Approx. 1.3 mA
Input circuit	Sink
Input resistance	18.4 kΩ
Additional functions	4 edge detection units with timestamp function, 4x input oversampling
Input frequency	40 kHz
Switching threshold	
Low	<5 VDC
High	>15 VDC
Insulation voltage between channel and bus	500 V_{eff}
Digital outputs	
Quantity	Up to 4, configuration as input or output using software
Variant	Push / Pull / Push-Pull
Nominal voltage	24 VDC
Switching voltage	24 VDC -15% / +20%
Nominal output current	0.1 A
Total nominal current	0.4 A
Output circuit	Sink and/or source
Output protection	Thermal shutdown in the event of overcurrent or short circuit, integrated protection for switching inductive loads
Diagnostic status	Output monitoring
Leakage current when the output is switched off	Max. 25 µA
$R_{DS(on)}$	150 mΩ
Residual voltage	<0.9 V at 0.1 A nominal current
Peak short-circuit current	<10 A
Switch-on in the event of overload shutdown or short-circuit shutdown	Approx. 10 ms (depends on the module temperature)
Switching delay	
0 → 1	<2 µs
1 → 0	<2 µs

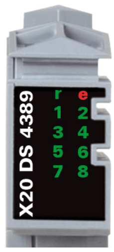
Table 2: X20DS4389 - Technical data

Order number	X20DS4389
Switching frequency	
Resistive load	Max. 24 kHz
Inductive load	See section "Switching inductive loads".
Braking voltage when switching off inductive loads	Switching voltage + 0.6 VDC
Insulation voltage between channel and bus	500 V_{eff}
Additional functions	4x microsecond-accurate edge generation, 4x output oversampling
Edge detection units	
Quantity	4
Operating mode	4 pulse duration measurements, relative or absolute moments of input edges in microsecond resolution, 4 history elements per unit
Counter size	16/32-bit
Input frequency (max.)	40 kHz
Resolution	125 ns timestamp function
Signal form	Square wave pulse
Sensor power supply	Module-internal, max. 600 mA
Edge generation units	
Quantity	4
Edge generation	
Absolute	Absolute to NetTime
Relative	Relative to other edges
Offset at relative edge generation	
Range of values	16-bit or 32-bit value
Resolution	1 µs
Actuator power supply	Module-internal, max. 600 mA
Oversampling	
Quantity	4
Sample time	25 to 255 µs
Data volume	Up to 64 bits per X2X Link cycle in the input and output direction
Electrical properties	
Electrical isolation	Channel isolated from bus Channel not isolated from channel
Operating conditions	
Mounting orientation	
Horizontal	Yes
Vertical	Yes
Installation elevation above sea level	
0 to 2000 m	No limitation
>2000 m	Reduction of ambient temperature by 0.5°C per 100 m
Degree of protection per EN 60529	IP20
Ambient conditions	
Temperature	
Operation	
Horizontal mounting orientation	-25 to 60°C
Vertical mounting orientation	-25 to 50°C
Derating	See section "Derating".
Storage	-40 to 85°C
Transport	-40 to 85°C
Relative humidity	
Operation	5 to 95%, non-condensing
Storage	5 to 95%, non-condensing
Transport	5 to 95%, non-condensing
Mechanical properties	
Note	Order 1x terminal block X20TB12 separately. Order 1x bus module X20BM11 separately.
Pitch	$12.5^{+0.2}$ mm

Table 2: X20DS4389 - Technical data

3.2 LED status indicators

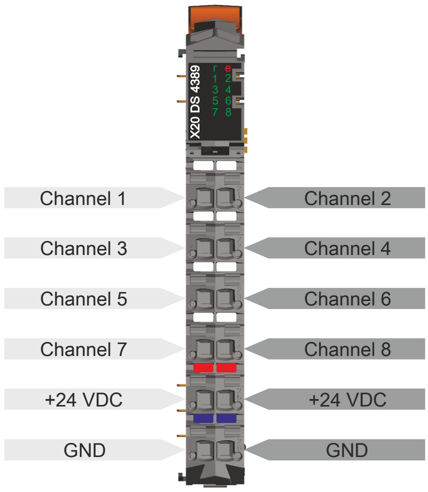
For a description of the various operating modes, see section "Additional information - Diagnostic LEDs" in the X20 system user's manual.

Figure	LED	Color	Status	Description
	r	Green	Off	No power to module
			Single flash	RESET mode
			Blinking	PREOPERATIONAL mode
			Double flash	Mode BOOT (during firmware update)[1]
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			On	Error or reset status
			Double flash	One of the following errors occurred: • Oversample output control error • Oversample output copy error • Edge detect poll cycle violation • Error on edge generator unit 1 - 4
	1 - 8	Green		Status of the corresponding digital signal

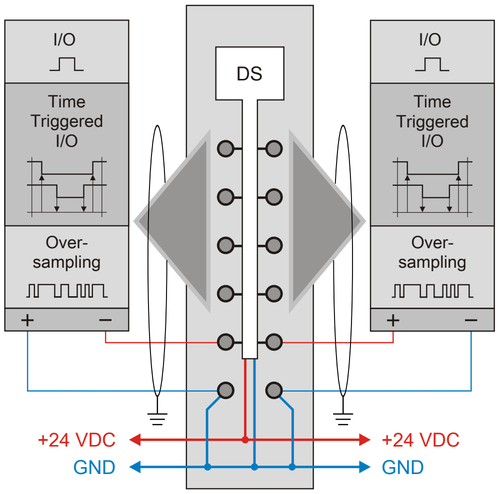
1) Depending on the configuration, a firmware update can take up to several minutes.

3.3 Pinout

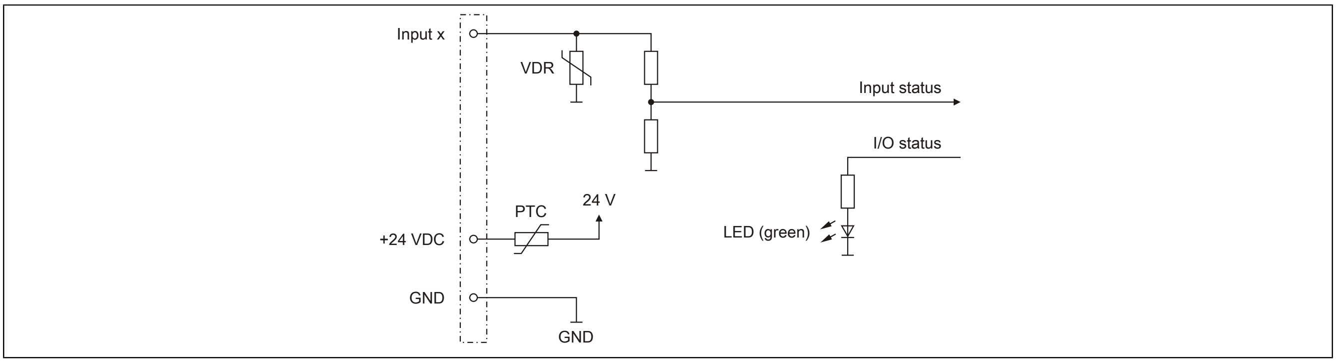
Shielded cables must be used for all signal lines.



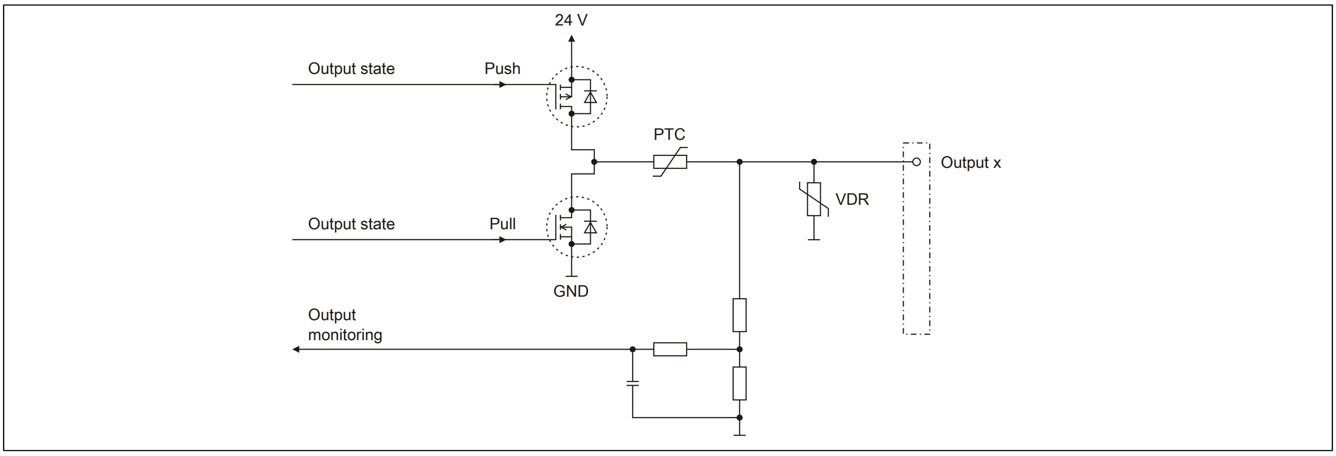
3.4 Connection example



3.5 Input circuit diagram



3.6 Output circuit diagram

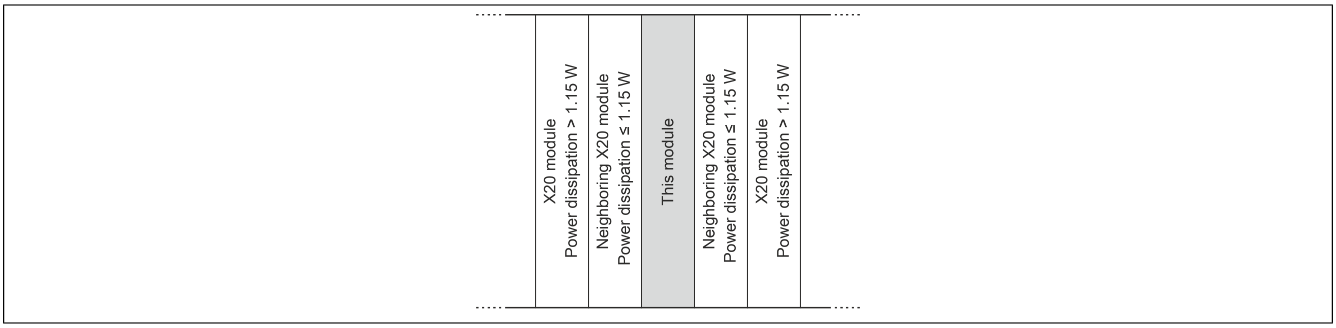


3.7 Derating

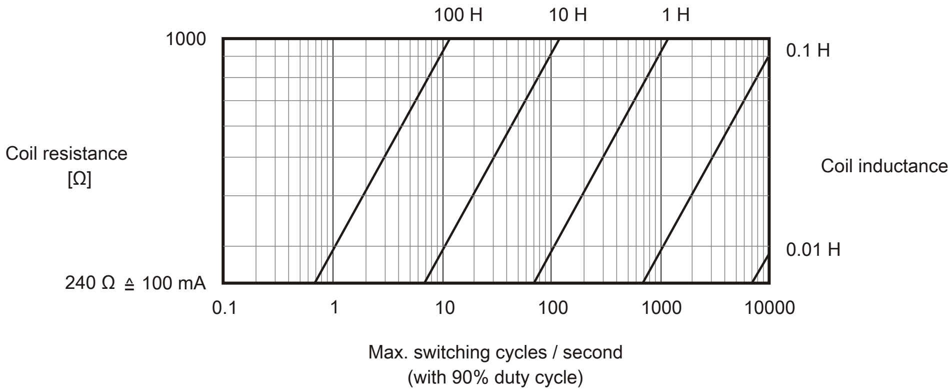
There is no derating when operated below 55°C.

During operation over 55°C, the power dissipation of the modules to the left and right of this module is not permitted to exceed 1.15 W!

For an example of calculating the power dissipation of I/O modules, see section "Mechanical and electrical configuration - Power dissipation of I/O modules" in the X20 user's manual.



3.8 Switching inductive loads



4 Register description

4.1 General data points

In addition to the registers described in the register description, the module has additional general data points. These are not module-specific but contain general information such as serial number and hardware variant.

General data points are described in section "Additional information - General data points" in the X20 system user's manual.

4.2 Function model 0 - default

Register	Name	Data type	Read		Write	
			Cyclic	Acyclic	Cyclic	Acyclic
Configuration - General						
513	CfO_SIframeGenID	USINT				•
Configuration - System timer						
642	CfO_SystemCycleTime	UINT				•
646	CfO_SystemCycleOffset	INT				•
650	CfO_SystemCyclePrescaler	UINT				•
Configuration - Physical I/O						
769 + (N-1) * 2	CfO_PhyIOConfigCh0N (Index N = 1 to 8)	USINT				•
Configuration - Direct I/O						
899	CfO_DirectIOClearMask0_7	USINT				•
903	CfO_DirectIOSetMask0_7	USINT				•
905	CfO_OutputUpdateCycle	USINT				•
Configuration - Oversampled I/O						
1025	CfO_OversampleMode	USINT				•
1027	CfO_OversampleSampleCycleID	USINT				•
1029	CfO_OversampleRelativeCycleID	USINT				•
1031	CfO_OversampleConsumeCycleID	USINT				•
1033	CfO_OversampleOutputBits	USINT				•
1035	CfO_OversampleInputBits	USINT				•
1037	CfO_OversampleOutputWindow	USINT				•
1039	CfO_OversampleInputWindow	USINT				•
1041 + (N*2)	CfO_OversampleConfigInputN (Index N = 0 to 3)	USINT				•
1049 + (N*2)	CfO_OversampleConfigOutputN (Index N = 0 to 3)	USINT				•
Configuration - Edge detection						
2817	CfO_EdgeDetectPollCycleID	USINT				•
2828	CfO_EdgeDetectEventEnable	UDINT				•
3073 + (N-1) * 16	CfO_EdgeDetectUnit0NMode (Index N = 1 to 4)	USINT				•
3075 + (N-1) * 16	CfO_EdgeDetectUnit0NLeading (Index N = 1 to 4)	USINT				•
3077 + (N-1) * 16	CfO_EdgeDetectUnit0NMaster (Index N = 1 to 4)	USINT				•
3079 + (N-1) * 16	CfO_EdgeDetectUnit0NSlave (Index N = 1 to 4)	USINT				•
Configuration - Edge generator						
2945	CfO_EdgeGenPollCycleEventID	USINT				•
2947	CfO_EdgeGenConsumeCycleEventID	USINT				•
3585 + (N-1) * 64	CfO_EdgeGenUnit0NMode (Index N = 1 to 4)	USINT				•
3589 + (N-1) * 64	CfO_EdgeGenUnit0NTimestampFifoLim (Index N = 1 to 4)	USINT				•
3591 + (N-1) * 64	CfO_EdgeGenUnit0NTimestampRegCount (Index N = 1 to 4)	USINT				•
3596 + (N-1) * 64	CfO_EdgeGenUnit0NPickupDiff	UDINT				•
3602 + (N-1) * 64	CfO_EdgeGenUnit0NConfigEdge0 (Index N = 1 to 4)	UINT				•
3606 + (N-1) * 64	CfO_EdgeGenUnit0NConfigEdge1 (Index N = 1 to 4)	UINT				•
3610 + (N-1) * 64	CfO_EdgeGenUnit0NConfigEdge2 (Index N = 1 to 4)	UINT				•
3614 + (N-1) * 64	CfO_EdgeGenUnit0NConfigEdge3 (Index N = 1 to 4)	UINT				•
Communication - General						
546	ProtocolError (16-bit)	UINT	•			
547	ProtocolError (8-bit)	USINT	•			
550	ProtocolSequenceViolation (16-bit)	UINT	•			
551	ProtocolSequenceViolation (8-bit)	USINT	•			
Communication - Error register						

Register	Name	Data type	Read		Write	
			Cyclic	Acyclic	Cyclic	Acyclic
257	Error state - Output data and edge detection	USINT	•			
	OutputControlError	Bit 4				
	OutputCopyError	Bit 5				
	EdgeDetectError	Bit 6				
259	Error messages - Edge generator	USINT	•			
	EdgeGen01Error	Bit 0				
	EdgeGen01Warning	Bit 1				
	EdgeGen02Error	Bit 2				
	EdgeGen02Warning	Bit 3				
	EdgeGen03Error	Bit 4				
	EdgeGen03Warning	Bit 5				
	EdgeGen04Error	Bit 6				
	EdgeGen04Warning	Bit 7				
321	Acknowledge error messages - Output data and edge detection	USINT			•	
	QuitOutputControlError	Bit 4				
	QuitOutputCopyError	Bit 5				
	QuitEdgeDetectError	Bit 6				
323	Acknowledge error messages - Edge generator	USINT			•	
	QuitEdgeGen01Error	Bit 0				
	QuitEdgeGen01Warning	Bit 1				
	QuitEdgeGen02Error	Bit 2				
	QuitEdgeGen02Warning	Bit 3				
	QuitEdgeGen03Error	Bit 4				
	QuitEdgeGen03Warning	Bit 5				
	QuitEdgeGen04Error	Bit 6				
	QuitEdgeGen04Warning	Bit 7				
Communication - System timer						
683	SDCLifeCount	SINT	•			
Communication - Direct I/O						
915	Output state	USINT			•	
	DigitalOutput03	Bit 2				
	DigitalOutput04	Bit 3				
	DigitalOutput07	Bit 6				
	DigitalOutput08	Bit 7				
927	Input state	USINT	•			
	DigitalInput01	Bit 0				
	...	...				
	DigitalInput08	Bit 7				
Communication - Oversampled I/O (output)						
1059	Oversampling configuration	USINT			•	
	OversampleEnable	Bit 0				
	OversampleOutputValidate	Bit 1				
1063	OversampleOutputCycle	USINT			•	
	OversampleSampleOffset	USINT			•	
1088 + N	OversampleOutput0NSample1_8 (Index N = 1 to 4)	USINT			•	
1092 + N	OversampleOutput0NSample9_16 (Index N = 1 to 4)	USINT			•	
1096 + N	OversampleOutput0NSample17_24 (Index N = 1 to 4)	USINT			•	
1100 + N	OversampleOutput0NSample25_32 (Index N = 1 to 4)	USINT			•	
1104 + N	OversampleOutput0NSample33_40 (Index N = 1 to 4)	USINT			•	
1108 + N	OversampleOutput0NSample41_48 (Index N = 1 to 4)	USINT			•	
1112 + N	OversampleOutput0NSample49_56 (Index N = 1 to 4)	USINT			•	
1116 + N	OversampleOutput0NSample57_64 (Index N = 1 to 4)	USINT			•	
Communication - Oversampled I/O (input)						
1074	OversampleInputTime	INT	•			
1079	OversampleInputCycle	USINT	•			
1120 + N	OversampleInput0NSample64_57 (Index N = 1 to 4)	USINT	•			
1124 + N	OversampleInput0NSample56_49 (Index N = 1 to 4)	USINT	•			
1128 + N	OversampleInput0NSample48_41 (Index N = 1 to 4)	USINT	•			
1132 + N	OversampleInput0NSample40_33 (Index N = 1 to 4)	USINT	•			
1136 + N	OversampleInput0NSample32_25 (Index N = 1 to 4)	USINT	•			
1140 + N	OversampleInput0NSample24_17 (Index N = 1 to 4)	USINT	•			
1144 + N	OversampleInput0NSample16_9 (Index N = 1 to 4)	USINT	•			
1148 + N	OversampleInput0NSample8_1 (Index N = 1 to 4)	USINT	•			
Communication - Edge detection						
4098 + (N-1) * 32	EdgeDetect0NMastercount (16-bit) (Index N = 1 to 4)	INT	•			
4099 + (N-1) * 32	EdgeDetect0NMastercount (8-bit) (Index N = 1 to 4)	SINT	•			
4102 + (N-1) * 32	EdgeDetect0NSlavecount (16-bit) (Index N = 1 to 4)	INT	•			
4103 + (N-1) * 32	EdgeDetect0NSlavecount (8-bit) (Index N = 1 to 4)	SINT	•			
4108 + (N-1) * 32	EdgeDetect0NDifference (32-bit) (Index N = 1 to 4)	DINT	•			

Register	Name	Data type	Read		Write	
			Cyclic	Acyclic	Cyclic	Acyclic
4110 + (N-1) * 32	EdgeDetect0NDifference (16-bit) (Index N = 1 to 4)	INT	•			
4116 + (N-1) * 32	EdgeDetect0NMastertime (32-bit) (Index N = 1 to 4)	DINT	•			
4118 + (N-1) * 32	EdgeDetect0NMastertime (16-bit) (Index N = 1 to 4)	INT	•			
4124 + (N-1) * 32	EdgeDetect0NSlavetime (32-bit) (Index N = 1 to 4)	DINT	•			
4126 + (N-1) * 32	EdgeDetect0NSlavetime (16-bit) (Index N = 1 to 4)	INT	•			
Communication - Edge generator						
6145 + (N-1) * 256	Enabling units	USINT			•	
	EdgeGen0NEnable EdgeGen0NEnableReadback (Index N = 1 to 4)	Bit 0				
6147 + (N-1) * 256	EdgeGenSequence	USINT			•	
	EdgeGenSequenceReadback	USINT	•			
6180 + (N-1) * 256	EdgeGen0NOffset1 (Index N = 1 to 4) (32-bit) CfO_EdgeGen0NOffset_32bit1 (Index N = 1 to 4)	UDINT			•	• (CfO_EdgeGen0NOffset_32bit1)
6182 + (N-1) * 256	EdgeGen0NOffset1 (Index N = 1 to 4) (16-bit)	UINT			•	
6188 + (N-1) * 256	EdgeGen0NOffset2 (Index N = 1 to 4) (32-bit) CfO_EdgeGen0NOffset_32bit2 (Index N = 1 to 4)	UDINT			•	•
6190 + (N-1) * 256	EdgeGen0NOffset2 (Index N = 1 to 4) (16-bit)	UINT			•	
6196 + (N-1) * 256	EdgeGen0NOffset3 (Index N = 1 to 4) (32-bit) CfO_EdgeGen0NOffset_32bit3 (Index N = 1 to 4)	UDINT			•	•
6198 + (N-1) * 256	EdgeGen0NOffset3 (Index N = 1 to 4) (16-bit)	UINT			•	
6204 + (N-1) * 256	EdgeGen0NOffset4 (Index N = 1 to 4) (32-bit) CfO_EdgeGen0NOffset_32bit4 (Index N = 1 to 4)	UDINT			•	•
6206 + (N-1) * 256	EdgeGen0NOffset4 (Index N = 1 to 4) (16-bit)	UINT			•	
6212 + (N-1) * 256	EdgeGen0NTimestamp1 (Index N = 1 to 4) (32-bit)	UDINT			•	
6214 + (N-1) * 256	EdgeGen0NTimestamp1 (Index N = 1 to 4) (16-bit)	UINT			•	
6220 + (N-1) * 256	EdgeGen0NTimestamp2 (Index N = 1 to 4) (32-bit)	UDINT			•	
6222 + (N-1) * 256	EdgeGen0NTimestamp2 (Index N = 1 to 4) (16-bit)	UINT			•	
6228 + (N-1) * 256	EdgeGen0NTimestamp3 (Index N = 1 to 4) (32-bit)	UDINT			•	
6230 + (N-1) * 256	EdgeGen0NTimestamp3 (Index N = 1 to 4) (16-bit)	UINT			•	
6236 + (N-1) * 256	EdgeGen0NTimestamp4 (Index N = 1 to 4) (32-bit)	UDINT			•	
6238 + (N-1) * 256	EdgeGen0NTimestamp4 (Index N = 1 to 4) (16-bit)	UINT			•	

4.3 General

4.3.1 Use with Automation Studio

The module is only supported by SG4 target systems via X2X and POWERLINK!

X2X Link supports the following synchronous cyclic data per module:

- 31 bytes input data consisting of 30 input bytes and X2X status byte
- 30 bytes output data

To optimize use and prevent needless data transfer, data points can be adjusted as needed in Automation Studio. Unnecessary data points can be disabled, and the bit width of the data points can be defined.

4.3.2 Timestamp function

The timestamp function is based on synchronized timers. If a timestamp event occurs, the module immediately saves the current NetTime. After the respective data is transferred to the CPU, including this precise moment, the CPU can then evaluate the data using its own NetTime (or system time), if necessary.

Conversely, the CPU can predefine output events, apply a timestamp and transfer them to the module. The module then executes the predefined action at the precise moment defined by the CPU.

For more information about NetTime and timestamps, see "NetTime Technology" on page 37.

The resolution of the timestamp is up to 1/8 µs in both directions.

4.3.2.1 Synchronization jitter

Because the CPU – which specifies the X2X NetTime – and the module have different clocks, the module's internal X2X NetTime must be synchronized with the CPU's NetTime. Due to this synchronization, the module's internal X2X NetTime is corrected by a maximum of 1/8 µs per system cycle if necessary. This synchronization jitter becomes noticeable when using the NetTime with 1/8 µs resolution (max. ±1/8 µs).

If a 100% exact 1/8 µs resolution without jitter is required, then the "localtime 1/8 µs" must be used (see register "CfO_EdgeDetectUnitMode" on page 27).

4.4 General registers

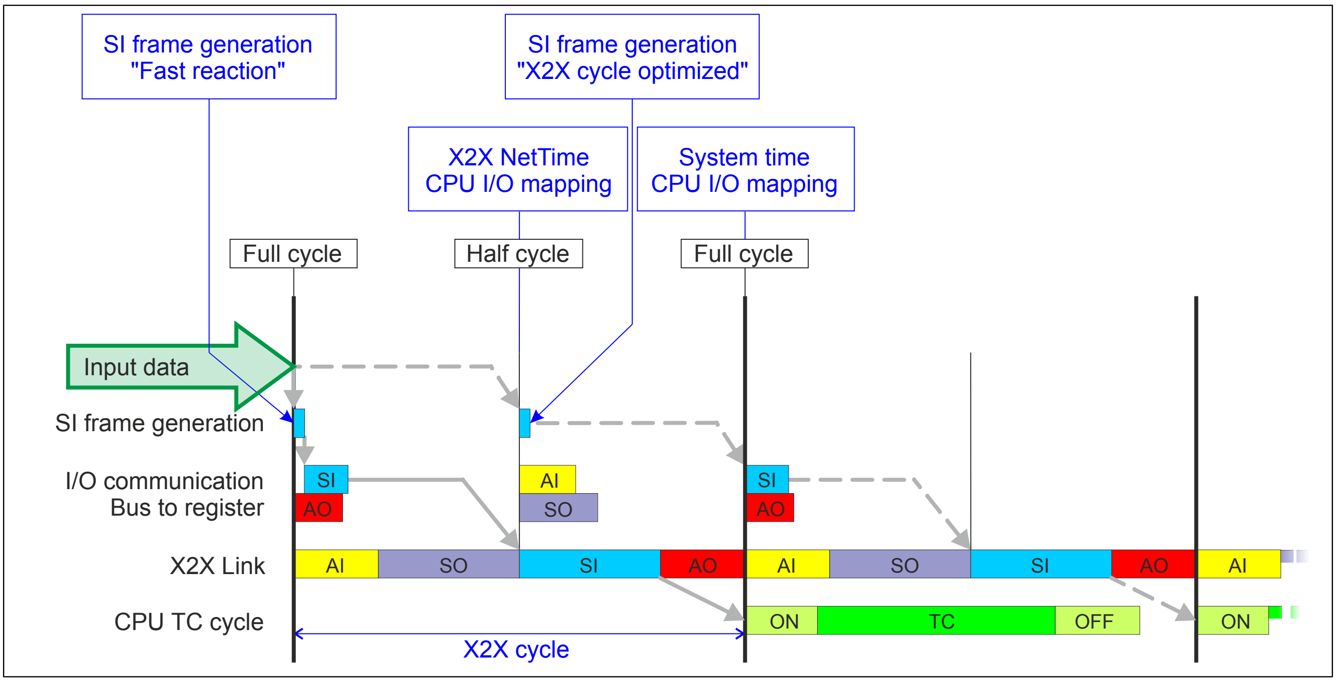
4.4.1 Defining the moment for generating synchronous input data

Name:
CfO_SIframeGenID
"SI frame generation" in the Automation Studio I/O configuration.

When the synchronous input data is generated for transfer is defined in this register. This has a decisive effect on the timing of the input data.

Setting "Fast reaction" causes the input data to be available one X2X cycle sooner in the CPU. However, this setting also has a negative effect on the minimum X2X cycle time.

Data type	Values	Information
USINT	9	X2X cycle optimized
	14	Fast reaction



4.4.2 Number of X2X protocol errors

Name:
ProtocolError

This register contains an error counter that specifies the number of X2X protocol errors. In the I/O configuration, parameter "Network information" can be used to help configure a data point for this register with a bit width of 8 or 16 bits in the I/O mapping.

Data type	Values	Information
USINT	0 to 255	Error counter (8-bit)
UINT	0 to 65535	Error counter (16-bit)

4.4.3 Number of X2X sequence violations

Name:
ProtocolSequenceViolation

This register contains an error counter that specifies the number of X2X sequence violations. In the I/O configuration, parameter "Network information" can be used to help configure a data point with a bit width of 8 or 16 bits in the I/O mapping.

Data type	Values	Information
USINT	0 to 255	Error counter (8-bit)
UINT	0 to 65535	Error counter (16-bit)

4.4.4 System clock counter for checking the validity of the data frame

Name:
SDCLifeCount

Counter that is incremented with each system timer cycle. "SDC information" in the Automation Studio I/O configuration can be used to enable this register in the I/O mapping as data point "SDCLifeCount".

The 8-bit counter register is needed for the SDC software package. It is incremented with the system clock to allow the SDC to check the validity of the data frame.

Data type	Values
SINT	-128 to 127

4.5 Error handling

If one of the functions detects an error, then an error bit is set in one of the error state registers. The application is now able to react to this and acknowledge the errors by setting a respective bit in the "Acknowledge error message" registers. This causes the bit to be reset in the error state register. If the error source persists, then the error bit is set again as soon as the error is detected again (i.e. resetting is not possible).

Error acknowledgment has no effect on the functionality of the module. The module resumes processing, automatically if possible, as soon as the error source is eliminated.

If an error occurs (not a warning), this is indicated by the red "e" LED on the module (double flash). This signal is automatically acknowledged as soon as the error source has been eliminated.

4.5.1 Error state - Output data and edge detection

Name:
OutputControlError
OutputCopyError
EdgeDetectError

Data output errors and cycle time setting errors are indicated in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 3	Reserved	-	
4	OutputControlError	0	No error
		1	The module did not receive new data in time when "Output control mode = Single", meaning that a bit that has already been output would have been output again by the output control buffer.
5	OutputCopyError	0	No error
		1	Oversampling output data could not be copied to the output control buffer (attempted to write to an address outside the oversample output window, for example).
6	EdgeDetectError	0	No error
		1	Edge detection cycle time violation: "EdgeDetectPollCycle" must be ≤255 µs. This error is occurs if the cycle set in register "CfO_EdgeDetectPollCycleID" on page 26 is >255 µs.
7	Reserved	-	

4.5.2 Error messages - Edge generator

Name:
EdgeGen01Error to EdgeGen04Error
EdgeGen01Warning to EdgeGen04Warning

This register indicates edge detection errors.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	EdgeGen01Error	0	No error
		1	Unit 1 error[1)]
1	EdgeGen01Warning	0	No error
		1	Unit 1 warning[2)]
2	EdgeGen02Error	0	No error
		1	Unit 2 error[1)]
3	EdgeGen02Warning	0	No error
		1	Unit 2 warning[2)]
4	EdgeGen03Error	0	No error
		1	Unit 3 error[1)]
5	EdgeGen03Warning	0	No error
		1	Unit 3 warning[2)]
6	EdgeGen04Error	0	No error
		1	Unit 4 error[1)]
7	EdgeGen04Warning	0	No error
		1	Unit 4 warning[2)]

1) **Possible errors**

- Due to "EdgeGenPollCycle", one or more timestamps from the edge generator of a unit were not able to be processed in time, and it was not possible to catch back up (see register "CfO_EdgeGenUnitPickupDiff" on page 33).
- A branched ring-shaped chain of edges in a unit is attempting to set the timestamp for an edge even though the FIFO buffer of the configured physical channel is already full (see register "CfO_EdgeGenUnitConfigEdge" on page 34 → Ring-shaped chain of edges).

2) Due to "EdgeGenPollCycle", one or more timestamps from the edge generator of a unit were not able to be processed in time, and it was possible to catch back up (see register "CfO_EdgeGenUnitPickupDiff" on page 33).

4.5.3 Acknowledging error messages - Output data and edge detection

Name:
QuitOutputControlError
QuitOutputCopyError
QuitEdgeDetectError

Error messages from register "Error state - Output data and edge detection" on page 12 can be acknowledged by setting the corresponding bits in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 3	Reserved	-	
4	QuitOutputControlError	0	No change
		1	Acknowledge error
5	QuitOutputCopyError	0	No change
		1	Acknowledge error
6	QuitEdgeDetectError	0	No change
		1	Acknowledge error
7	Reserved	-	

4.5.4 Acknowledge error messages - Edge generator

Name:
QuitEdgeGen01Error to QuitEdgeGen04Error
QuitEdgeGen01Warning to QuitEdgeGen04Warning

The error message from register "Error messages - Edge generator" on page 13 can be acknowledged in this register by setting the respective bit.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	QuitEdgeGen01Error	0	No change
		1	Acknowledge error
1	QuitEdgeGen01Warning	0	No change
		1	Acknowledge warning
2	QuitEdgeGen02Error	0	No change
		1	Acknowledge error
3	QuitEdgeGen02Warning	0	No change
		1	Acknowledge warning
4	QuitEdgeGen03Error	0	No change
		1	Acknowledge error
5	QuitEdgeGen03Warning	0	No change
		1	Acknowledge warning
6	QuitEdgeGen04Error	0	No change
		1	Acknowledge error
7	QuitEdgeGen04Warning	0	No change
		1	Acknowledge warning

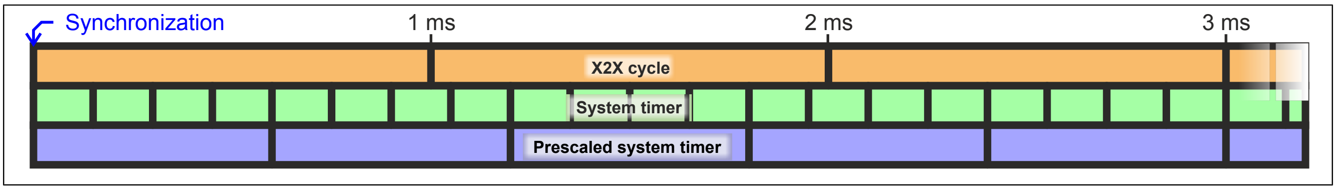
4.6 System timer

The module's individual functions all depend on a system timer. This internal "system cycle time" can be set from 25 to 255 µs. The functions can also be run using a configurable "prescaled system timer" to minimize the load on the module, thereby making it possible to use the shortest possible X2X cycle time.

The cycle of the "prescaled system timer" (and system timer) is referenced with the X2X Link as soon as the module has been started up and the X2X Link has been initialized. Since the system timer and the module's internal NetTime use the same clock, the two run synchronously from that point on. An X2X cycle time that is not a multiple of the system cycle time results in an offset, which can be calculated, however.

The following values apply to the following example:

X2X cycle	1 ms
System timer	150 µs
Prescaled system timer	4



4.6.1 Setting the cycle time of the system timer

Name:
CfO_SystemCycleTime
"Cycle time" in the Automation Studio I/O configuration.

The cycle time of the system timer can be set in steps of 1/8 µs in this register. The value entered in the Automation Studio I/O configuration is automatically multiplied by 8.

Information:

A setting <50 µs has a negative effect on the minimum X2X cycle time!

Data type	Values	Information
UINT	200 to 2047	System timer cycle time in steps of 1/8 µs (25 to 255.875 µs)

4.6.2 Offsetting the synchronization moment of the system cycle

Name:
CfO_SystemCycleOffset
"Cycle offset" in the Automation Studio I/O configuration.

The synchronization moment for the system cycle can be offset in steps of 1/8 µs in this register. The value entered in the Automation Studio I/O configuration is automatically multiplied by 8.

Data type	Values	Information
INT	-32768 to 32767	Cycle offset in steps of 1/8 µs (-4096 to 4095.875 µs)

4.6.3 Configuration of the cycle prescaler

Name:
CfO_SystemCyclePrescaler
"Cycle prescaler" in the Automation Studio I/O configuration.

The prescaler for setting the prescaled system timer can be configured in this register. The cycle time of the specified system timer is a product of the system timer multiple set in this register.

The "prescaled system timer" can be used as an alternative time source for the individual functions. This is useful if a function requires a very short system cycle. To reduce the load on the module in such a situation, other functions can be processed in a slow cycle.

Data type	Values	Information
UINT	2 to 128	Multiple of the system timer

4.7 Physical I/O configuration

4.7.1 "CfO_PhyIOConfigCh" registers

Name:
CfO_PhyIOConfigCh01 to CfO_PhyIOConfigCh08

The physical I/O channels can each be configured individually in these registers.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	Push driver[1)	0	Disabled
		1	Enabled
1	Pull driver[1)	0	Disabled
		1	Enabled
2	Input inverted	0	Not inverted
		1	Inverse
3	Output inverted[1)	0	Not inverted
		1	Inverse
4 - 7	Output function[1)	0	Direct I/O
		1 to 15	Reserved

1) Only available for the I/O channels 3, 4, 7 and 8.

4.8 Direct I/O

Direct I/O makes it possible to use the physical I/Os like normal I/Os. Additionally, the application can only set or reset I/Os (e.g. an output channel is set by the edge generator and manually reset by the application).

4.8.1 Direct operation of the output channel - Reset

Name:
CfO_DirectIOClearMask0_7
"Direct operation of output channel 03" to "Direct operation of output channel 08" in the Automation Studio I/O configuration.

If the bit for the respective channel is set in this register, then the output is reset as soon as its direct I/O output channel (register "DigitalOutput" on page 17 or "DigitalOutput0x" in the Automation Studio I/O mapping) is reset.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 1	Reserved	-	
2	Output channel 3	0	No change
		1	Reset channel
3	Output channel 4	0	No change
		1	Reset channel
4 - 5	Reserved	-	
6	Output channel 7	0	No change
		1	Reset channel
7	Output channel 8	0	No change
		1	Reset channel

4.8.2 Direct operation of the output channel - Set

Name:
CfO_DirectIOSetMask0_7
"Direct operation of output channel 03" to "Direct operation of output channel 08" in the Automation Studio I/O configuration.

If the bit for the respective channel is set in this register, then the output is set as soon as its direct I/O output channel (register "DigitalOutput" on page 17 or "DigitalOutput0x" in the Automation Studio I/O mapping) is set.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 1	Reserved	-	
2	Output channel 3	0	No change
		1	Set channel
3	Output channel 4	0	No change
		1	Set channel
4 - 5	Reserved	-	
6	Output channel 7	0	No change
		1	Set channel
7	Output channel 8	0	No change
		1	Set channel

4.8.3 Direct operation of the output channel - Moment of data output

Name:
CfO_OutputUpdateCycle

The moment when data is output is set with this register.

Data type	Values	Information
USINT	10	X2X cycle optimized (jitter-free)
	15	Fast reaction (with jitter)

4.8.4 Output state

Name:
DigitalOutput03 and DigitalOutput04, DigitalOutput07 and DigitalOutput08

This register contains the bits for controlling the direct I/O output channels. Depending on the configuration of registers "CfO_DirectIOClearMask0_7" on page 16 and "CfO_DirectIOSetMask0_7" on page 16, the digital outputs are set to the status of the respective bit in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 1	Reserved	-	
2	DigitalOutput03	0 or 1	Output state of channel 3
3	DigitalOutput04	0 or 1	Output state of channel 4
4 - 5	Reserved	-	
6	DigitalOutput07	0 or 1	Output state of channel 7
7	DigitalOutput08	0 or 1	Output state of channel 8

4.8.5 Input state

Name:
DigitalInput01 to DigitalInput08

The state of the digital input channels is contained in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	DigitalInput01	0 or 1	Input state of channel 1
...	...	...	
7	DigitalInput08	0 or 1	Input state of channel 8

4.9 Oversampled I/O

"Oversampled I/O" is based on input status buffers and output control buffers. The input data acquisition and output control occur in one sample cycle (one sample cycle corresponds to one bit in the buffer). The precise moment of an input buffer entry is indicated by its position in the buffer and the NetTime assigned to the buffer.

When "Output control mode = Single", every output buffer entry is marked as invalid once it has been executed. This ensures that the outputs are not supplied with invalid data. In this mode, the application needs to ensure that the module is always supplied with valid data.

When using "Output control mode = Continuous" the contents of the buffer are output again if the module is not supplied with new oversample output data.

4.9.1 Addressing the output control buffer

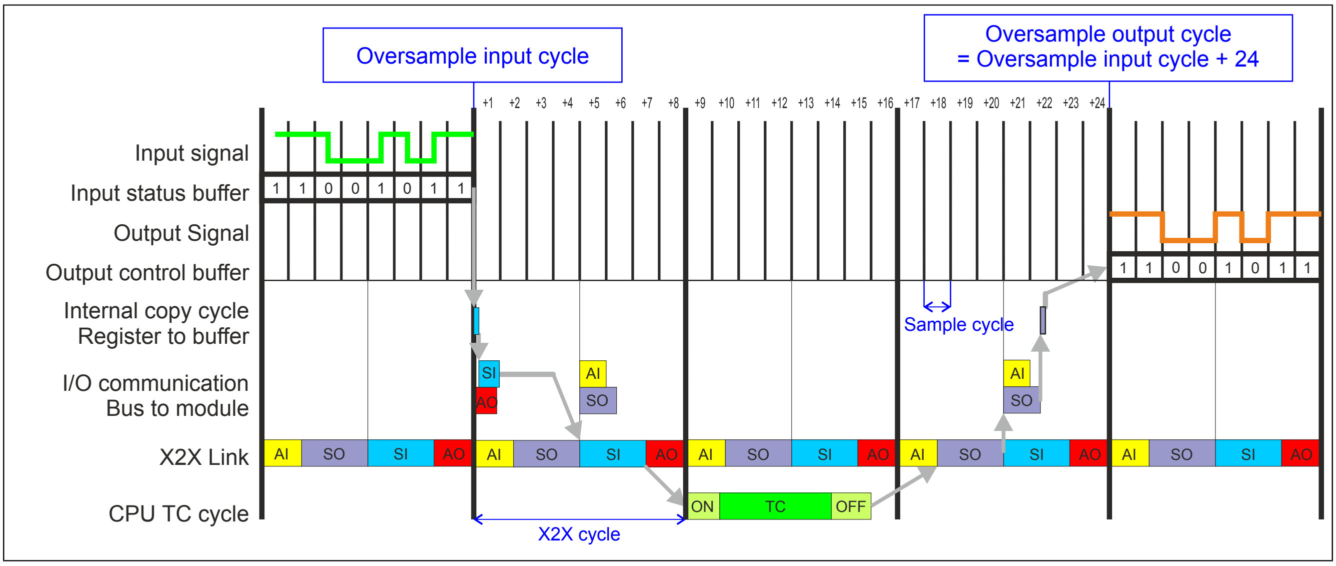
The module has one cyclic 256-bit output control buffer for each oversample channel. One bit is output from these buffers to the configured physical output channels in each "sample cycle". When new data is transferred to one of these buffers, the application must define where in the respective buffer the data should be written to. There are 2 possibilities available for this (absolute or relative "Output mode" in the Automation Studio I/O configuration).

4.9.2 Absolute addressing of the output control buffer

With absolute addressing, in each cycle where "OversampleOutputValidate = True", in addition to the oversample output sample data (in the "OversampleOutput0NSample" on page 24 registers) an address must also be transferred in register "OversampleOutputCycle" on page 24. This address defines where in the output control buffer the new data should be copied. In order to calculate this address, the contents of register "OversampleInputCycle" on page 25, which contains the address of the most recently output data, and the transfer time to the module must be taken into account. To help avoid incorrect addressing of the output control buffer, the buffer section that is capable of being written to can be limited using register "OversampleOutputWindow" on page 22. This window will always be shifted relative to the current sample address. An "OutputCopyError" will be triggered if an attempt is made to write to an address that is outside of this window.

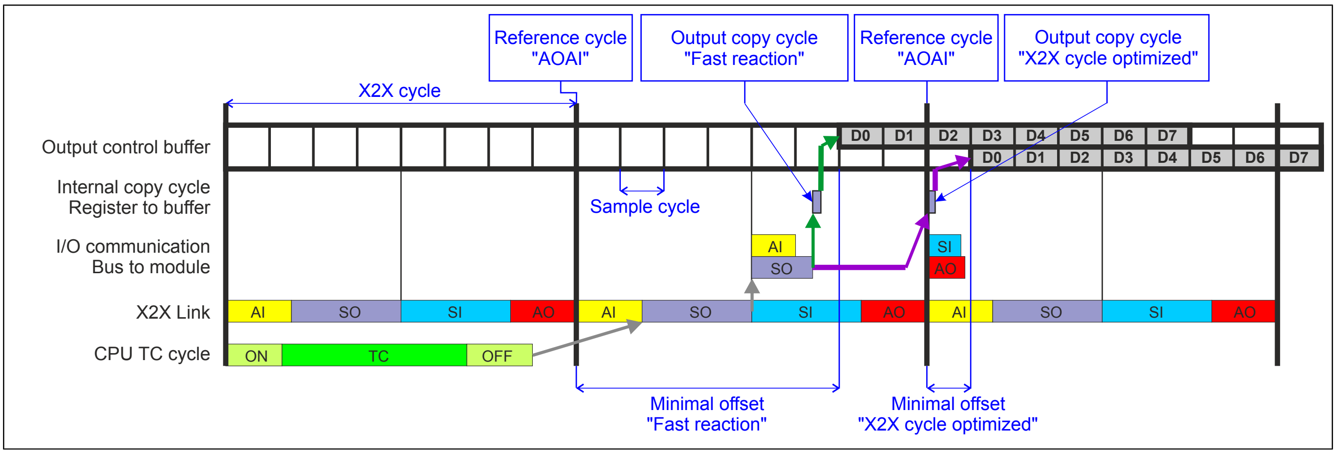
Example

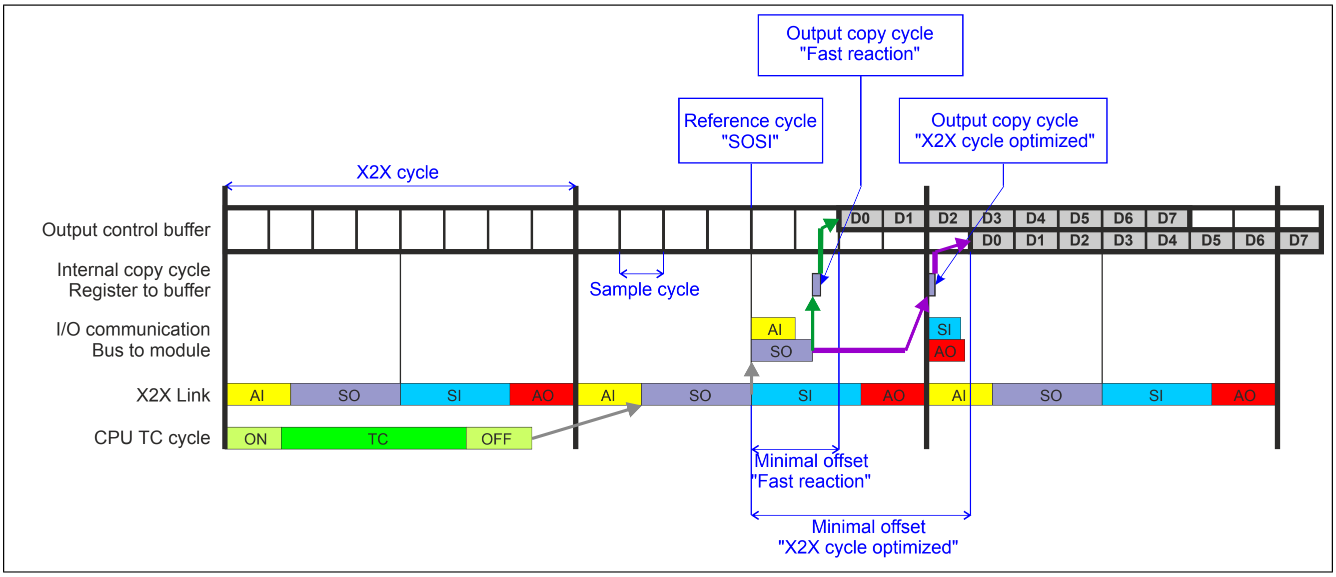
Timing characteristics from the oversample input cycle to the oversample output cycle in absolute output mode ("SI frame generation = Fast reaction", "Output copy cycle = Fast reaction", 8 samples per X2X cycle):



4.9.3 Relative addressing of the output control buffer

When "OversampleOutputValidate = True", then the oversample output sample data is automatically copied to an address relative to the last referenced address at the set output copy cycle moment. Register "OversampleSampleOffset" on page 24 serves as the offset. The new data cannot start being output immediately at the output copy cycle moment because it takes time to copy the data from the registers to the buffer. This means that an offset of 0 is not allowed. The relative output control buffer address + offset must point to an address within the "oversample output window". The oversample output window is always offset relative to the current sample address. An OutputCopyError is triggered if an attempt is made to write to an address that is outside of this window.





4.9.4 Configuration of the output control buffer

Name:
CfO_OversampleMode
"Output mode" in the Automation Studio I/O configuration
"Output control mode" in the Automation Studio I/O configuration

The output control buffer can be configured globally for all channels in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	Addressing the output control buffer "Output mode"	0	Absolute addressing of the output control buffer
		1	Relative addressing of the output control buffer
1	Cyclic output control "Output control mode"	0	Single - Output control buffer entry is marked invalid after execution.
		1	Continuous - Output control buffer entry is not changed.
2 - 7	Reserved	-	

Cyclic output control

If cyclic output control is enabled, then all data in the output control buffer is marked invalid as soon as it is output ("Output control mode = Single"). An OutputControlError is generated if the module does not receive data in time, thereby causing a situation in which a bit that has already been output would be output in the buffer again. In this type of error situation, the output takes on the "Output default state" configured in register "CfO_OversampleConfigOutput" on page 23.

If cyclic output control is disabled, then the data is output again if the output control buffer overflows ("Output control mode = Continuous").

Information:

All 256 bits of the output control buffer are always output.

4.9.5 Configuration of the source for the sample cycle

Name:
CfO_OversampleSampleCycleID
"Sample cycle" in the Automation Studio I/O configuration.

The source for the sample cycle is configured in this register. During each sample cycle, one bit from the output control buffers of the oversampled I/O channels is output to the configured physical output, and the status of the configured inputs is read into one bit of the respective input status buffer.

Data type	Values	Information
USINT	2	System timer The value set in register "CfO_SystemCycleTime" on page 14 is used as the sample cycle.
	3	Prescaled system timer The "prescaled system timer" is used as sample cycle.
	10	AOAI The sample cycle is clocked with the AOAI interrupt of the X2X cycle.
	14	SOSI The sample cycle is clocked with the SOSI interrupt of the X2X cycle.

4.9.6 Configuration of the source for the user interface reference cycle

Name:
CfO_OversampleRelativeCycleID
"Reference cycle" in the Automation Studio I/O configuration.

The source for the user interface reference cycle is configured in this register.

- The input data is referenced at the moment of the reference cycle. The referenced data is then copied to the "oversample input sample register" on page 25 at the moment of SI frame generation, taking into account the oversample input window.
- With relative addressing of the output control buffer, the new sample data is copied to an address relative to the output control buffer address current to the "reference cycle".
- The reference cycle is also used to reference the sample cycle and thus the output data production and input data acquisition (e.g. to the X2X cycle).

Data type	Values	Information
USINT	2	System timer The value set in register "CfO_SystemCycleTime" on page 14 is used as the reference cycle.
	3	Prescaled system timer The prescaled system timer is used as sample cycle.
	10	AOAI The sample cycle is referenced with the AOAI interrupt of the X2X cycle.
	14	SOSI The sample cycle is referenced with the SOSI interrupt of the X2X cycle.

4.9.7 Defining the moment for copying the data to the output control buffer

Name:
CfO_OversampleConsumeCycleID
"Output copy cycle" in the Automation Studio I/O configuration.

At the time of the output copy cycle, data is copied from the "OversampleOutput0NSample" on page 24 registers into the output control buffer.

When "Output copy cycle = Fast reaction", it is not possible to determine when the data is copied to the output control buffer in either of the two addressing modes. The copy cycles will experience a certain degree of jitter depending on the module load. However, this only affects the moment of the internal copy procedures and therefore the moment of the earliest possible output sample. This will not affect the quality of the output signal. However, "Output copy cycle = Fast reaction" also has a negative effect on the minimum X2X cycle time.

When using the setting "Output copy cycle = X2X cycle optimized", be aware that the sample data cannot start being output immediately at the "Output copy cycle" time due to the internal copy cycle to the output control buffers.

Data type	Values	Information
USINT	10	X2X cycle optimized The output data is copied to the output control buffer with the AOAI interrupt of the X2X cycle.
	15	Fast reaction The output data is copied to the output control buffer immediately after being received.

4.9.8 Number of output bits to be transferred

Name:
CfO_OversampleOutputBits
"User interface size" in the Automation Studio I/O configuration.

Specifies how many bits are transferred from the "OversampleOutput0NSample" on page 24 registers to the output control buffers at the output copy cycle moment.

Data type	Values	Information
USINT	1 to 64	Output bits

4.9.9 Number of input bits to be transferred

Name:
CfO_OversampleInputBits
"User interface size" in the Automation Studio I/O configuration.

Specifies how many bits are transferred from the input status buffer to the "OversampleInput0NSample" on page 25 registers during SI frame generation.

Data type	Values	Information
USINT	1 to 64	Input bits

4.9.10 Write area in the output control buffer

Name:
CfO_OversampleOutputWindow
"Output control mode" in the Automation Studio I/O configuration.

Defines the area in the output control buffer to which data is permitted to be written. The window is always offset relative to the current sample position. (a value of 128, for example, means that the 128 bits following the current sample cycle can be written to). An OutputCopyError is triggered if an attempt is made to write output sample data to a location outside of this window.
In Automation Studio, the value for this register is set to 128 bits with "Output control mode = Single" and to 255 bits with "Output control mode = Continuous".

Data type	Values	Information
USINT	0 to 255	Output window

4.9.11 Defining the moment for referencing input data

Name:
CfO_OversampleInputWindow
"Input mode" in the Automation Studio I/O configuration.

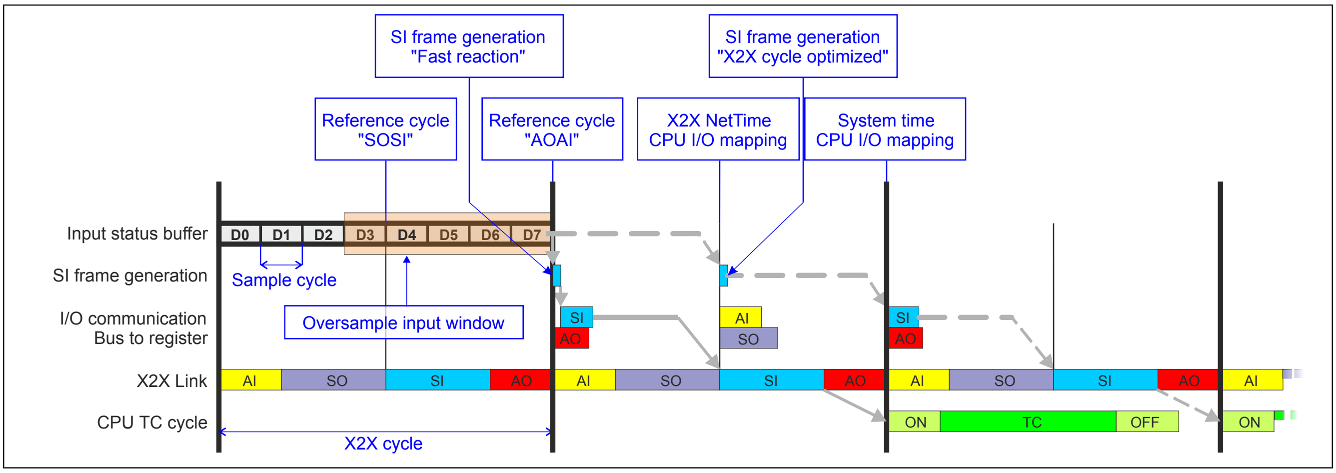
The "oversample input window" defines when the input data is referenced. It is located chronologically before SI frame generation. If the reference moment ("reference cycle" on page 21) is within this window, then the referenced data from the input status buffer is copied to register "OversampleInput0NSample" on page 25. If the moment at which the reference occurs is outside the "oversample input window" then the data that is most recent at the moment of "SI frame generation" is copied from the input status buffer to register "OversampleInput0NSample" on page 25.
This register is limited internally to the value from register "CfO_OversampleInputBits" on page 21.

Information:

As a result, the oversample input time and oversample input cycle are set either at the reference time or at the moment of "SI frame generation".

In Automation Studio, the value for this register is set to 63 with "Input mode = Referenced values" and to 0 with "Input mode = Most recent values".

Data type	Values	Information
USINT	0 to 63	Input window



4.9.12 Assigning between the physical input channel and oversample I/O input

Name:
CfO_OversampleConfigInput
"Oversample I/O 01 → Input" to "Oversample I/O 04 input" in the Automation Studio I/O configuration.

Which physical input channel an oversample I/O input should be linked to is defined in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 3	Number of the physical input channel	0	Input channel 1
		..	
		7	Input channel 8
4 - 7	Reserved	-	

4.9.13 Configuring the outputs of the oversampling channels

Name:
CfO_OversampleConfigOutput
"Oversample I/O 01 → Output" to "Oversample I/O 04 → Output" in the Automation Studio I/O configuration
"Oversample I/O 01 → Output control" to "Oversample I/O 04 → Output control" in the Automation Studio I/O configuration
"Oversample I/O 01 → Output default value" to "Oversample I/O 04 → Output default value" in the Automation Studio I/O configuration

This register helps configure the outputs of the individual oversample channels.

The "Output default state" bits define which level the respective output takes on before oversampling is started. In addition, the output is set to the defined "Output default state" in the event of an error.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 3	Number of the physical output channel "Oversample I/O 0x → Output"	2	Output channel 3
		3	Output channel 4
		6	Output channel 7
		7	Output channel 8
4	Output: Clear "Oversample I/O 0x → Output control"	0	Output cannot be reset by the oversample channel.
		1	Output can be reset by the oversample channel.
5	Output: Set "Oversample I/O 0x → Output control"	0	Output cannot be set by the oversample channel.
		1	Output can be set by the oversample channel.
6	Default output state: Clear "Oversample I/O 0x → Output default value"	0	Output not cleared by default
		1	Output cleared by default
7	Default output state: Set "Oversample I/O 0x → Output default value"	0	Output not set by default
		1	Output set by default

4.9.14 Oversampling configuration

Name:
OversampleEnable
OversampleOutputValidate

The oversampling and copy process for the output buffer can be configured in this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	OversampleEnable	0	Disables oversampling (with the next reference cycle)
		1	Enables oversampling (with the next reference cycle)
1	OversampleOutputValidate	0	Disable the copy procedure to the output control buffer.
		1	Enables the copy procedure to the output control buffer. • Used to synchronize the oversampling procedure at startup. • This makes it possible to prevent new data from being transferred to the "OversampleOutput0NSample" on page 24 registers in each X2X cycle.
2 - 7	Reserved	-	

4.9.15 Address of the new output sampling data in the output control buffer

Name:
OversampleOutputCycle

When absolute addressing of the output control buffer is being used, this register specifies the address from which the new output sample data should be copied to the output control buffer.

Data type	Values	Information
USINT	0 to 255	Address of the output control buffer

4.9.16 Offset of new output sample data

Name:
OversampleSampleOffset

When relative addressing of the output control buffer is being used, this register serves as the offset for the new output sample data. (Sample address at the time of the reference cycle + Offset = Address to which the new output sample data is copied in the output control buffer).

Data type	Values	Information
USINT	0 to 255	Offset of output sample data

4.9.17 Oversample output sample data

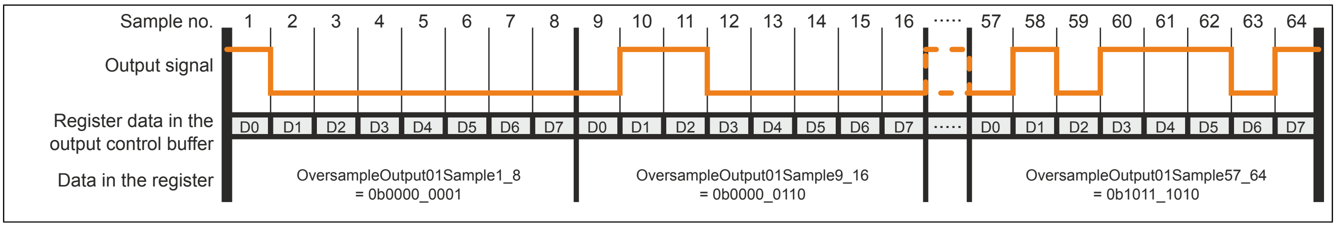
Name:
OversampleOutput01Sample1_8 to OversampleOutput04Sample1_8
OversampleOutput01Sample9_16 to OversampleOutput04Sample9_16
OversampleOutput01Sample17_24 to OversampleOutput04Sample17_24
OversampleOutput01Sample25_32 to OversampleOutput04Sample25_32
OversampleOutput01Sample33_40 to OversampleOutput04Sample33_40
OversampleOutput01Sample41_48 to OversampleOutput04Sample41_48
OversampleOutput01Sample49_56 to OversampleOutput04Sample49_56
OversampleOutput01Sample57_64 to OversampleOutput04Sample57_64

Contains the oversample output sample data. Up to 64 samples (8 bytes) for each oversample I/O channel can be synchronously transferred with a X2X cycle. This data is copied to the specified address (absolute or relative) in the output control buffer at the set output copy cycle. 1 bit of this data is then output during each "sample cycle" to the physical output that is assigned to the oversample I/O channel.
Bit 0 of "OversampleOutputSample1_8" is copied to the output control buffer first, meaning that it is the first bit that is output. "OversampleOutputSample57_64" bit 7 is the last bit to be output.

Data type	Values	Information
USINT	0 to 255	Output sample data

Example

Assignment of "OversampleOutputSample" register data to output signal



4.9.18 X2X NetTime of the input data

Name:
OversampleInputTime

This register contains the 2 low-order bytes of the X2X NetTime from the moment at which the oversample input data was referenced. This provides an easy way to accurately calculate the moment of each individual input sample.

For more information about NetTime and timestamps, see "NetTime Technology" on page 37.

Data type	Values	Information
INT	-32768 to 32767	X2X NetTime of the input data in microseconds

4.9.19 Input status buffer address of the input sample data

Name:
OversampleInputCycle

This register contains the input status buffer address of the input sample data.
In addition, the value in this register can be used to reference an absolute addressing of the output control buffer.

Data type	Values	Information
USINT	0 to 255	Input status buffer address

4.9.20 Input sample data

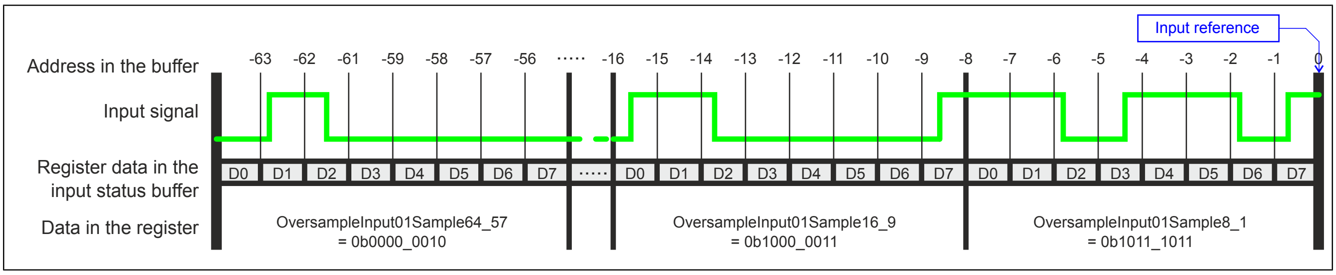
Name:
OversampleInput01Sample8_1 to OversampleInput04Sample8_1
OversampleInput01Sample16_9 to OversampleInput04Sample16_9
OversampleInput01Sample24_17 to OversampleInput04Sample24_17
OversampleInput01Sample32_25 to OversampleInput04Sample32_25
OversampleInput01Sample40_33 to OversampleInput04Sample40_33
OversampleInput01Sample48_41 to OversampleInput04Sample48_41
OversampleInput01Sample56_49 to OversampleInput04Sample56_49
OversampleInput01Sample64_57 to OversampleInput04Sample64_57

The data of the 4 oversample input status buffers are copied to this register at the moment of SI frame generation. A maximum of 64 samples (8 bytes) per oversample I/O channel can be synchronously retrieved from the oversample input status buffer with each X2X cycle.
The most recent input sample bit is stored in "OversampleInputSample8_1" bit 7. The oldest input sample is stored in "OversampleInputSample64_57" bit 0.

Data type	Values	Information
USINT	0 to 255	Input sample data

Example

Input signal and resulting data in "OversampleInputSample"



4.10 Edge detection

The module's edge detection function allows edges to be measured with microsecond precision. The concept is based on a maximum of 4 units. One master and one slave edge can be configured for each unit.

At the moment of each master edge, the NetTime of the master edge and the NetTime of a previous slave edge (if present) are logged. A "master counter" and a "slave counter" can always be used to determine how many edges have been detected since the last X2X cycle.

For timestamps and counters, the module has a history memory that can store up to 4 elements per unit. This means that several edges can also be measured accurately within one X2X cycle.

4.10.1 Configuring the source for the polling cycle

Name:
CfO_EdgeDetectPollCycleID
"Polling cycle" in the Automation Studio I/O configuration.

The source for the polling cycle can be configured in this register.

Information:

The polling cycle must be ≤255 µs. If the configured cycle >255 µs, EdgeDetectError occurs.

Data type	Values	Information
USINT	2	System timer The time set in register "CfO_SystemCycleTime" on page 14 is used for the polling cycle.
	3	Prescaled system timer The time set in register "CfO_SystemCyclePrescaler" on page 15 is used for the polling cycle.

4.10.2 Edge detection mode

Name:
CfO_EdgeDetectEventEnable
"Edge detection mode" in the Automation Studio I/O configuration.

The bits in this register define on which edges of the individual input channels an interrupt should be triggered for edge detection.

In mode "Event-triggered", the NetTime of each edge is recorded as an interrupt immediately when the edge occurs. However, an extremely large amount of interrupts within a short amount of time can prevent the module from being able to process any other operations in time!.

In mode "Polling", only the NetTime of the first edge that occurs within a polling cycle is recorded. This ensures that the module is not overloaded by too many edges.

In the Automation Studio I/O configuration, this register is initialized with 0x00000000 when "Edge detection mode = Polling" and with 0xFFFFFFFF when "Edge detection mode = Event-triggered".

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	Physical input 1	0	No interrupt triggered on falling edge
		1	Interrupt triggered on falling edge
...		...	
7	Physical input 8	0	No interrupt triggered on falling edge
		1	Interrupt triggered on falling edge
8 - 15	Reserved	-	
16	Physical input 1	0	No interrupt triggered on rising edge
		1	Interrupt triggered on rising edge
...		...	
23	Physical input 8	0	No interrupt triggered on rising edge
		1	Interrupt triggered on rising edge
24 - 31	Reserved	-	

4.10.3 Setting the time base, slave edge and master edge

Name:
CfO_EdgeDetectUnit01Mode to CfO_EdgeDetectUnit04Mode
"Time base" in the Automation Studio I/O configuration
"Slave edge" in the Automation Studio I/O configuration
"Master edge" in the Automation Studio I/O configuration

When using a time base with 1/8 µs resolution, keep in mind that the timestamps produced also have a resolution of exactly 1/8 µs. The respective conversions must be made for calculating in combination with the CPU system time or X2X NetTime.

In addition, synchronization jitter also plays a role when using "Time base = Nettime resolution 1/8 usec" (see "Synchronization jitter" on page 10). This means that exactly identical input edges can cause slight differences in the results. If 100% exact 1/8 µs resolution is required, then "Local resolution 1/8 usec" must be used.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 1	"Time base"	0	Local time 1/8 µs (Automation Studio: Local resolution 1/8 usec)
		1	Local time 1 µs (Automation Studio: Local resolution 1 usec)
		2	NetTime 1/8 µs (Automation Studio: Nettime resolution 1/8 usec)
		3	NetTime 1 µs (Automation Studio: Nettime resolution 1 usec)
2 - 5	Reserved	-	
6	"Slave edge"	0	Disabled
		1	Enabled
7	"Master edge"	0	Disabled
		1	Enabled

4.10.4 "CfO_EdgeDetectUnitLeading" register

Name:
CfO_EdgeDetectUnit01Leading to CfO_EdgeDetectUnit04Leading
"Slave leading" in the Automation Studio I/O configuration.

When a slave edge occurs, the current NetTime is always saved within the module. A FIFO buffer is provided inside the module that always stores the last 16 slave stamps (even when a master edge occurs).

This value determines from which position the slave time should be retrieved from the FIFO when a master edge occurs. This can be used to measure average periodic signals over multiple cycles.

Data type	Value	Information
USINT	0 to 15	Position in the slave edge FIFO

4.10.5 Source of the master edge per edge detection unit

Name:
CfO_EdgeDetectUnit01Master to CfO_EdgeDetectUnit01Master
"Master edge" in the Automation Studio I/O configuration.

The source of the master edge for the respective "edge detection unit" is defined in this register.

Data type	Values	Information
USINT	0	Rising edge on physical input 1
	...	...
	7	Rising edge on physical input 8
	16	Falling edge on physical input 1
	...	...
	23	Falling edge on physical input 8

4.10.6 Source of the slave edge per edge detection unit

Name:
CfO_EdgeDetectUnit01Slave to CfO_EdgeDetectUnit04Slave
"Slave edge" in the Automation Studio I/O configuration.

The source of the slave edge for the respective "edge detection unit" is defined in this register.

Data type	Values	Information
USINT	0	Rising edge on physical input 1
	...	...
	7	Rising edge on physical input 8
	16	Falling edge on physical input 1
	...	...
	23	Falling edge on physical input 8

4.10.7 "EdgeDetectSlavecount" register

Name:
EdgeDetect01Slavecount to EdgeDetect04Slavecount

The reference pulses of the detected slave edges are counted continuously in this register. The contents of this register are only updated when a master edge occurs. Up to 4 history elements can be enabled for these counters in the Automation Studio I/O configuration. These counters can detect if multiple slave edges occur before a master edge.

Data type	Value	Information
SINT	-128 to 127	Number of detected slave edges (8-bit)
INT	-32,768 to 32,767	Number of detected slave edges (16-bit)

4.10.8 "EdgeDetectDifference" register

Name:
EdgeDetect01Difference to EdgeDetect04Difference

Contains the time difference between a master edge and the last slave edge addressed via "Slave leading".

Data type	Value	Information
INT	-32,768 to 32,767	Time difference between master/slave edge (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Time difference between master/slave edge (32-bit)

4.10.9 Number of detected master edges

Name:
EdgeDetect01Mastercount to EdgeDetect04Mastercount

Detected master edges are counted in this register.

Data type	Values	Information
SINT	-128 to 127	Number of detected master edges (8-bit)
INT	-32768 to 32767	Number of detected master edges (16-bit)

4.10.10 NetTime when a master edge occurs

Name:
EdgeDetect01Mastertime to EdgeDetect04Mastertime

The exact NetTime is copied to this register when a master edge occurs.

For more information about NetTime and timestamps, see "NetTime Technology" on page 37.

Data type	Values	Information
INT	-32768 to 32767	NetTime master edge in microseconds (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	NetTime master edge in microseconds (32-bit)

4.10.11 "EdgeDetectSlavetime" register

Name:
EdgeDetect01Slavetime to EdgeDetect04Slavetime

In dieses Register wird beim Auftreten einer Masterflanke die exakte NetTime einer eventuell vorher aufgetretenen und durch "Slavevorlauf" adressierten, Slaveflanke kopiert. Pro Masterflanke kann nur eine Slavetime aus dem "Slavevorlauf FIFO" geholt werden. Das Auftreten mehrerer Flanken vor einer Masterflanke kann also nur durch den "EdgeDetectSlavecount" festgestellt werden.

For more information about NetTime and timestamps, see "NetTime Technology" on page 37.

Data type	Values	Information
INT	-32768 to 32767	NetTime slave edge in microseconds (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	NetTime slave edge in microseconds (32-bit)

History:

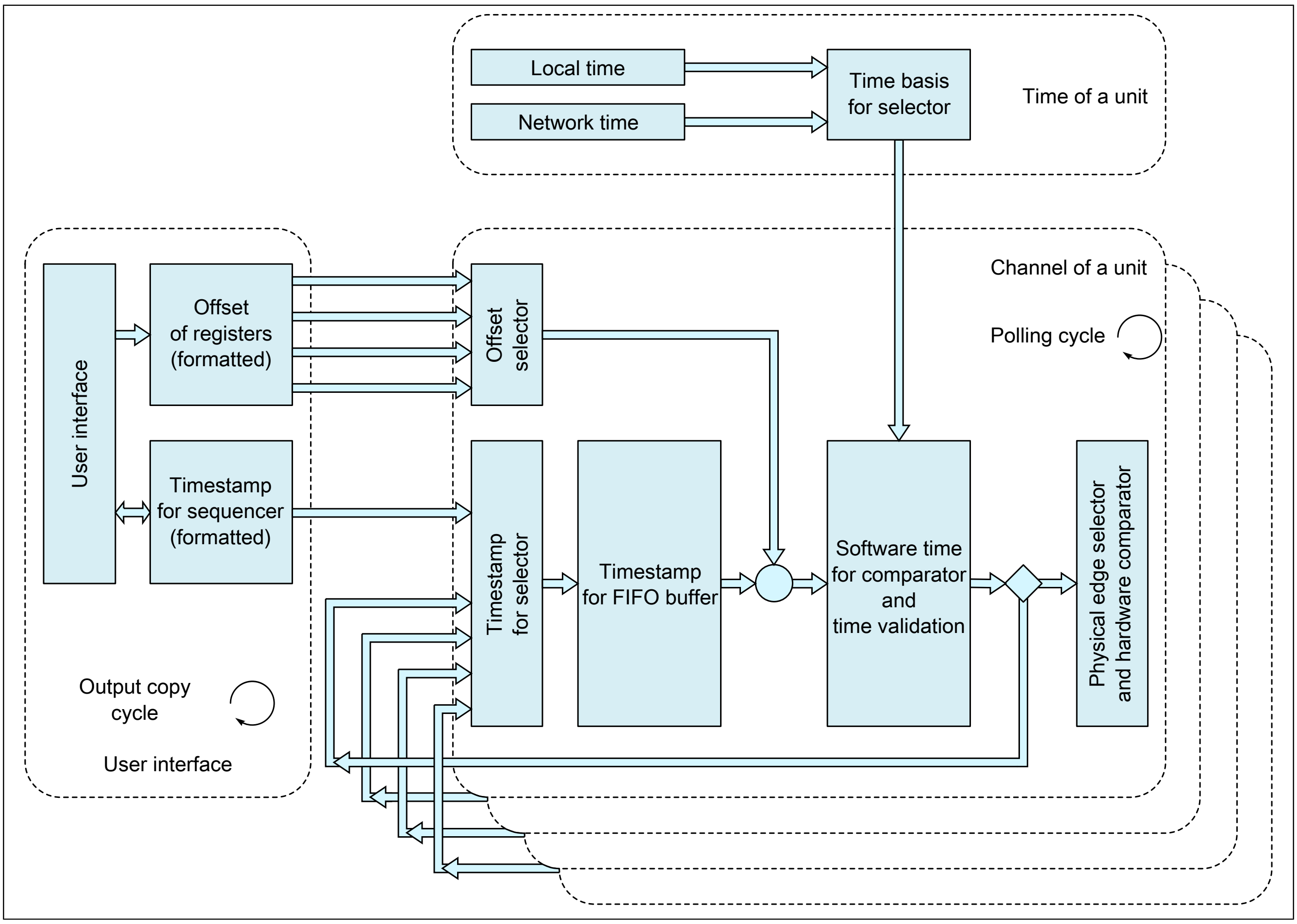
In Automation Studio, a history of up to 4 elements can be enabled in the I/O configuration for the following registers: "EdgeDetectSlavecount" on page 28, "EdgeDetectDifference" on page 28, "EdgeDetectMastertime" on page 28 and "EdgeDetectSlavetime" on page 29. Configured history elements are all transferred synchronously with each X2X cycle. This makes it possible to measure multiple edges precisely within a single X2X cycle.

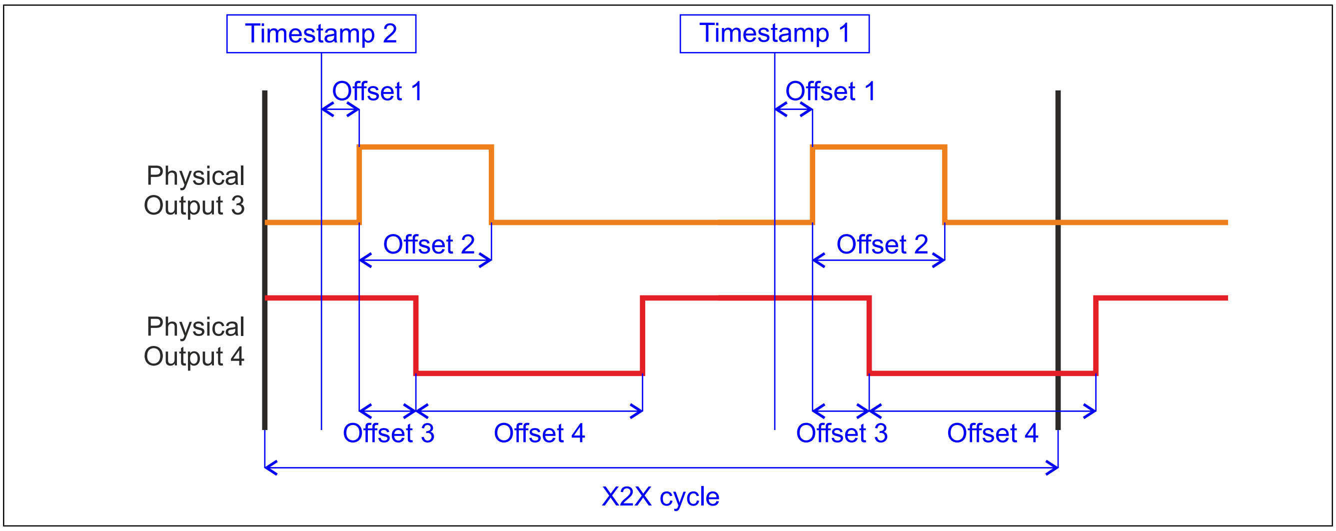
Information:

When the history is enabled, the maximum number of data bytes (28 bytes) that can be transferred synchronously via the X2X Link is reached quickly (especially if 32-bit data points are used).

4.11 Edge generator

The edge generator is based on 4 units. The units are able to generate edges independently of the X2X cycle. For each unit, up to 4 timestamps can be set per X2X cycle. The individual edges can then be referenced to this timestamp or to other edges using an offset.





4.11.1 Mode "DigitalCamSwitch"

"Unit 0x" in Automation Studio I/O configuration.

Starting with upgrade 1.1.0.2, "DigitalCamSwitch" mode can also be selected for each unit when configuring the edge generator in Automation Studio.
In this mode, the entire configuration and operation take place exclusively with the function blocks from the "ASMcDcs" motion library. For more information, see the corresponding ASMcDcs function block descriptions.

4.11.2 Preparing data for edge generation by hardware comparators

Name:
CfO_EdgeGenPollCycleEventID
"Generation cycle" in the Automation Studio I/O configuration.

To ensure edge output with microsecond precision, edge generation is based on internal hardware components. One such comparator is available for one rising and one falling edge respectively for each physical output channel. The data for the comparators is prepared in "EdgeGenPollCycle". Therefore, a maximum of one rising and one falling edge can be generated for each physical output channel per "EdgeGenPollCycle". If timestamps are set that cannot be processed in time due to this limitation, then an EdgeGenWarning is generated. Processing of such timestamps is made up for as quickly as possible, as long as they are within EdgeGenUnitPickupDiff.
The shorter this "generation cycle" is selected, the more negatively an enabled edge generator function affects the minimum X2X cycle time.

Data type	Value	Information
USINT	2	System timer
	3	Prescaled system timer

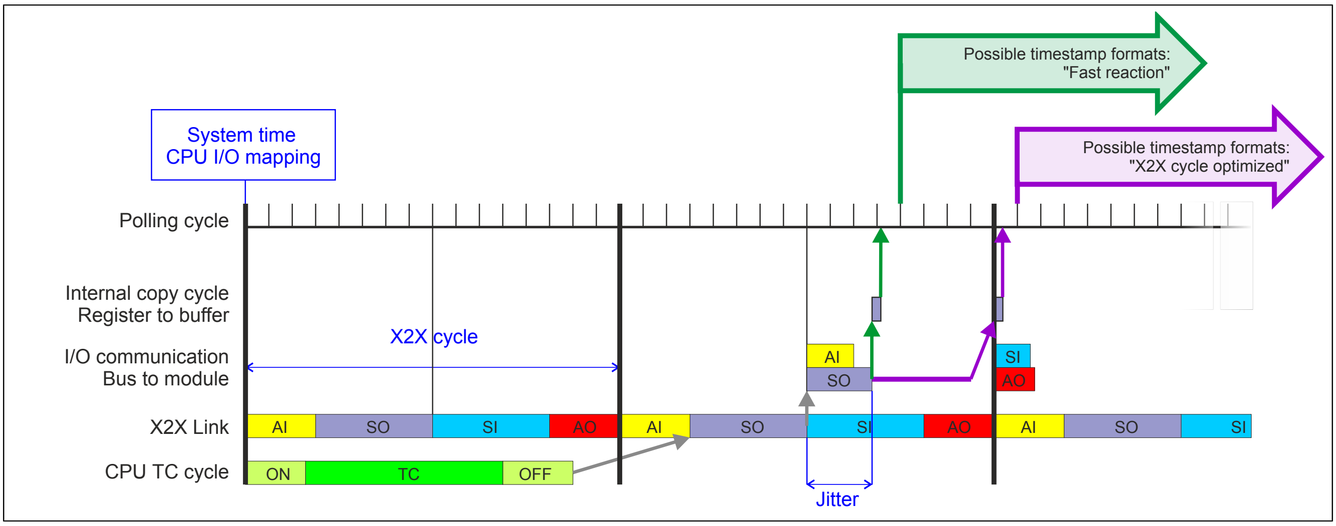
4.11.3 Moment when output data is applied for edge generation

Name:
CfO_EdgeGenConsumeCycleEventID

This register determines when the output data for edge generation is applied within the X2X cycle.

Data type	Value	Information
USINT	10	"X2X cycle optimized" The data is force-applied between the periods ASYNC IN (AI) and ASYNC OUT (AO).
	15	"Fast reaction (with jitter)" The data is applied immediately after SYNC OUT (SO) processing.

Setting "Fast reaction" results in jitter because the copy cycle of the SYNC OUT data can take different amounts of time. However, this only affects the moment at which the internal copy cycle takes place and therefore possibly also the earliest possible timestamp. Timestamps that are set outside of this jitter range are not affected by this.



4.11.4 Configuration of units

Name:
CfO_EdgeGenUnit01Mode to CfO_EdgeGenUnit04Mode
"Time base" in the Automation Studio I/O configuration
"Timestamp format" in the Automation Studio I/O configuration
"Offset format" in the Automation Studio I/O configuration
"Unit 01" to "Unit 04" in the Automation Studio I/O configuration

These registers contain the configuration bits for the respective units.

If "Timestamp resolution = 1/8 µs" is used, it is important to ensure that the timestamp data also has a resolution of 1/8 µs. Because the CPU system time and the X2X NetTime only have µs resolution, the system time or the NetTime must be offset by 3 bits to the left or multiplied by 8 in the application. This value can then be used as reference for timestamps with a resolution of 1/8 µs. It is also possible to use 1/8 µs timestamps from input edges as a reference.

For more information about NetTime and timestamps, see "NetTime Technology" on page 37.

When using the NetTime with 1/8 µs resolution, the synchronization jitter affects the output results (see "Synchronization jitter" on page 10).

Because the local time is not synchronized with the CPU system time or the X2X NetTime, this can only be used effectively together with a time source from the module (e.g. input edge timestamp on "local time").

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	Resolution of the timestamp "Time base" in the Automation Studio I/O configuration.	0	1 µs
		1	1/8 µs
1	Number of bits in the timestamp register "Timestamp format" in the Automation Studio I/O configuration.	0	16-bit
		1	32-bit
2	Offset resolution "Time base" in the Automation Studio I/O configuration.	0	1 µs
		1	1/8 µs
3	Number of bits in the offset register "Offset format" in the Automation Studio I/O configuration.	0	16-bit
		1	32-bit
4	Time base "Time base" in the Automation Studio I/O configuration.	0	NetTime
		1	Local time
5 - 6	Reserved	-	
7	Enable/disable units "Unit 0x" in the Automation Studio I/O configuration.	0	Disabled
		1	Enabled

4.11.5 Number of timestamps for FIFO

Name:
CfO_EdgeGenUnit01TimestampFifoLim to CfO_EdgeGenUnit04TimestampFifoLim

These registers define how many timestamps can be transferred to the FIFO buffer of a unit. The FIFO buffer serves as a memory buffer for timestamps in the future. Timestamps must be entered in the FIFO buffer in the same order in which they should be output. This means it is not possible to set a timestamp in the future followed by an earlier timestamp. The "EdgeGenSequenceReadback" on page 35 register can be used to indicate if the defined limit has been reached.

For more information about NetTime and timestamps, see "NetTime Technology" on page 37.

Data type	Value	Information
USINT	1 to 12	FIFO limit

4.11.6 Number of timestamps per X2X cycle

Name:
CfO_EdgeGenUnit01TimestampRegCount to CfO_EdgeGenUnit04TimestampRegCount
"Timestamp elements" in the Automation Studio I/O configuration.

This register determines how many timestamps can be transferred per X2X cycle.

For more information about NetTime and timestamps, see "NetTime Technology" on page 37.

Data type	Value	Information
USINT	1 to 4	Number of timestamps per X2X cycle

4.11.7 Pickup difference to be regained for timestamps

Name:
CfO_EdgeGenUnit01PickupDiff to CfO_EdgeGenUnit04PickupDiff

These registers are used to define how far in the past timestamps are permitted to be so that they are still caught up. Timestamps in the past are processed as quickly as possible as long as they are within the catch-up difference specified in this register. EdgeGenWarning is triggered as soon as a timestamp could not be processed in time and had to be "caught up". If a timestamp could not be caught up because it is outside the catch-up difference, "EdgeGenError" results in addition to the "EdgeGenWarning".

For more information about NetTime and timestamps, see "NetTime Technology" on page 37.

In Automation Studio, if "Timestamp format = 16-bit" this register is initialized with 65535 (0xFFFF), and if "Timestamp format = 32-bit" it is initialized with 134,217,728 (0x8000000).

Data type	Value	Information
UDINT	0 to 65535	Difference to be regained in µs when "Offset format = 16-bit"
	0 to 134,217,728	Difference to be regained in µs when "Offset format = 32-bit"

4.11.8 "CfO_EdgeGenUnitConfigEdge" register

Name:
CfO_EdgeGenUnit01ConfigEdge to CfO_EdgeGenUnit04ConfigEdge
"Unit 01→ Edge" to "Unit 04→ Edge" in the Automation Studio I/O configuration.
"Unit 01→ Mode" to "Unit 04→ Mode" in the Automation Studio I/O configuration.
"Unit 01→ Offset" to "Unit 04→ Offset" in the Automation Studio I/O configuration.
"Unit 01→ Unit 01" to "Unit 04→ Unit 04" in the Automation Studio I/O configuration.

The properties of each of the 4 edges of a unit can be configured in this register.

Ring-shaped chain of edges:

If the individual edges are linked together in a ring shape (e.g. edge 2 is relative to edge 1 and edge 1 is relative to edge 2) then a header must be determined for the ring using bit 11 "ring head enable" bit so that this type of cycle can start without timestamp. In Automation Studio, the "ring head enable" bit (bit 11) is set for edge 1 in all units by default. If this type of ring is branched (e.g. a third edge is relative to an edge within the ring) then you must make sure that the internal FIFO, which is available to every physical I/O edge, does not overflow. This happens if more than 12 edges are created by the ring, but should not be output until much later. If this situation occurs, whereby a ring creates edges even though the FIFO is full, then an EdgeGenError error is generated.

Data type	Values
UINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 4	Physical edge "Unit 0x →Edge" in the Automation Studio I/O configuration.	2	Channel 3 rising edge
		3	Channel 4 rising edge
		6	Channel 7 rising edge
		7	Channel 8 rising edge
		18	Channel 3 falling edge
		19	Channel 4 falling edge
		22	Channel 7 falling edge
		23	Channel 8 falling edge
5 - 7	Reserved	-	
8 - 10	Timestamp of FIFO source "Unit 0x →Mode" in the Automation Studio I/O configuration.	0	User interface, absolute
		1 to 3	Reserved
		4	Edge 1, relative
		5	Edge 2, relative
		6	Edge 3, relative
		7	Edge 4, relative
11	Ring-shaped chain Default in Automation Studio for "Edge 01 = 1", "Edge 02 = 0", "Edge 03 = 0", "Edge 04 = 0"	0	Disabled
		1	Enabled
12 -13	Offset register numbers "Unit 0x →Offset" in the Automation Studio I/O configuration.	0	Offset register 0
		1	Offset register 1
		2	Offset register 2
		3	Offset register 3
14	Reserved	-	
15	Switch edge on/off. "Unit 0x →Unit 0x" in the Automation Studio I/O configuration.	0	Disabled
		1	Enabled

4.11.9 Enabling units

Name:
EdgeGen01Enable to EdgeGen04Enable
EdgeGen01EnableReadback to EdgeGen04EnableReadback
"Unit 01" to "Unit 04" in the Automation Studio I/O configuration

The different units of the edge generator can be enabled/disabled using this register.

Data type	Values
USINT	See the bit structure.

Bit structure:

Bit	Description	Value	Information
0	EdgeGen0NEnable EdgeGen0NEnableReadback	0	Disabled
		1	Enabled
1 - 7	Reserved	-	

4.11.10 Sequence number for generating switching edges

Name:
EdgeGen01Sequence to EdgeGen04Sequence

If new timestamp data is to be applied to the module, then the sequence number must be increased by the number of timestamp elements that must be applied. If multiple elements are transferred within one X2X cycle, then you must make sure that the individual timestamps are placed in the FIFO buffer in the same order in which they occur chronologically. Data from EdgeGenTimestamp arrives in the FIFO buffer first; data from "EdgeGenTimestamp1" arrives last.

Data type	Value	Information
SINT	-128 to 127	Sequence number for generating switching edges

4.11.11 Last sequence number applied by the module for edge generation.

Name:
EdgeGen01SequenceReadback to EdgeGen04SequenceReadback

The sequence number is read back in this register. Like register "EdgeGenSequence" on page 35, this register is incremented if the specified timestamps can also be recorded by the module. If the module is not able to record any new timestamps (e.g. because EdgeGenUnitTimestampFifoLim has been reached), then this register indicates the number of the last sequence recorded by the module.

Data type	Value	Information
SINT	-128 to 127	Last sequence number accepted by the module for edge generation.

4.11.12 Offset formats

Automation Studio provides 3 different parameters for setting offsets.

- **Offset format:** This parameter makes it possible to select the file type (16-bit or 32-bit) for cyclic transfer and only affects "EdgeGenOffset" on page 35 registers.
  Acyclic transfer of offset values with register "CfO_EdgeGenOffset_32bit" on page 36 is not affected by this parameter and always remains 32 bits wide.
- **Offset 01 to Offset 04:** These parameters have 2 possible settings:
  - Initial configuration: The offset value is only written once during configuration.
  - Cyclic data: A data point is created in the Automation Studio I/O mapping and the offset value is written cyclically.
- **Offset 01 value to Offset 04 value:** The actual offset value.

4.11.12.1 "EdgeGenOffset" register

Name:
EdgeGen01Offset1 to EdgeGen04Offset1
...
EdgeGen01Offset4 to EdgeGen04Offset4
"Offset 01 value" to "Offset 04 value" in the Automation Studio I/O configuration

The 4 offsets of an edge generator unit are written in this register. Depending on the configuration in register "Edge generator unit mode" on page 32, the offset values are handled in µs or 1/8 µs steps.

For information regarding how to use the register and set the offset formats in Automation Studio, see "Offset formats" on page 35.

Data type	Value	Information
UINT	0 to 65535	16-bit offset
UDINT	0 to 134217728	Offset when "Offset format = 32-bit" and "Time base" = 1 µs
	0 to 1,073,741,824	Offset when "Offset format = 32-bit" and "Time base" = 1/8 µs

4.11.12.2 "CfO_EdgeGenOffset_32bit" register

Name:
CfO_EdgeGen01Offset_32bit1 to CfO_EdgeGen04Offset_32bit1
...
CfO_EdgeGen01Offset_32bit4 to CfO_EdgeGen04Offset_32bit4

The 4 offsets of an edge generator unit can be written acyclically using these registers. Depending on the configuration in register "Edge generator unit mode" on page 32, the offset values are handled in µs or 1/8 µs steps.

For information regarding how to use the register and set the offset formats in Automation Studio, see "Offset formats" on page 35.

Data type	Value	Information
UDINT	0 to 134217728	Offset when "Offset format = 32-bit" and "Time base" = 1 µs
	0 to 1,073,741,824	Offset when "Offset format = 32-bit" and "Time base" = 1/8 µs

4.11.13 Timestamp registers

Name:
EdgeGen01Timestamp1 to EdgeGen04Timestamp1
...
EdgeGen01Timestamp4 to EdgeGen04Timestamp4

Registers for the timestamps to which edges pending generation are referenced. Up to 4 timestamp elements can be transferred per X2X cycle. Between 1 and 4 of these timestamp elements are placed in the FIFO, depending on how much the sequence number is increased by. If an attempt is made to set timestamps to a time that has already passed, then EdgeGenWarning is generated (see register "CfO_EdgeGenUnitPickupDiff" on page 33).

For more information about NetTime and timestamps, see "NetTime Technology" on page 37.

Data type	Value
INT	-32768 to 32767
DINT	-2,147,483,648 to 2,147,483,647

4.12 NetTime Technology

NetTime refers to the ability to precisely synchronize and transfer system times between individual components of the controller or network (CPU, I/O modules, X2X Link, POWERLINK, etc.).

This allows the moment that events occur to be determined system-wide with microsecond precision. Upcoming events can also be executed precisely at a specified moment.

4.12.1 Time information

Various time information is available in the controller or on the network:

- System time (on the PLC, Automation PC, etc.)
- X2X Link time (for each X2X Link network)
- POWERLINK time (for each POWERLINK network)
- Time data points of I/O modules

The NetTime is based on 32-bit counters, which are increased with microsecond resolution. The sign of the time information changes after 35 min, 47 s, 483 ms and 648 µs; an overflow occurs after 71 min, 34 s, 967 ms and 296 µs.

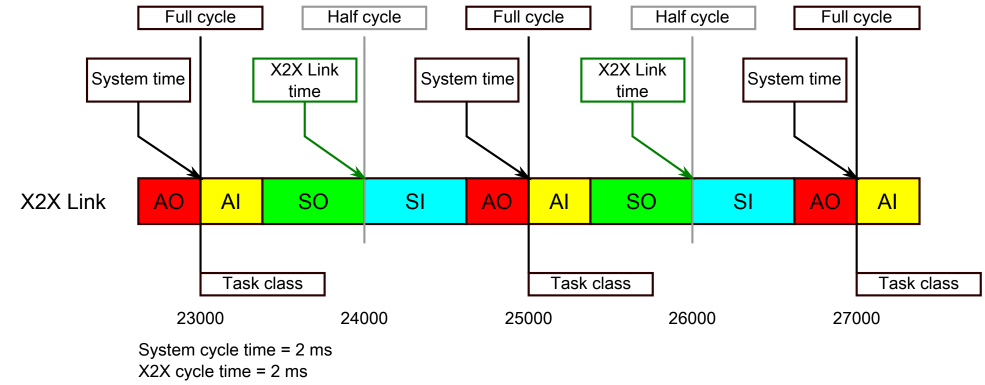
The initialization of the times is based on the system time during the startup of the X2X Link, the I/O modules or the POWERLINK interface.

Current time information in the application can also be determined via library AsIOTime.

4.12.1.1 PLC/Controller data points

The NetTime I/O data points of the PLC or the controller are latched to each system clock and made available.

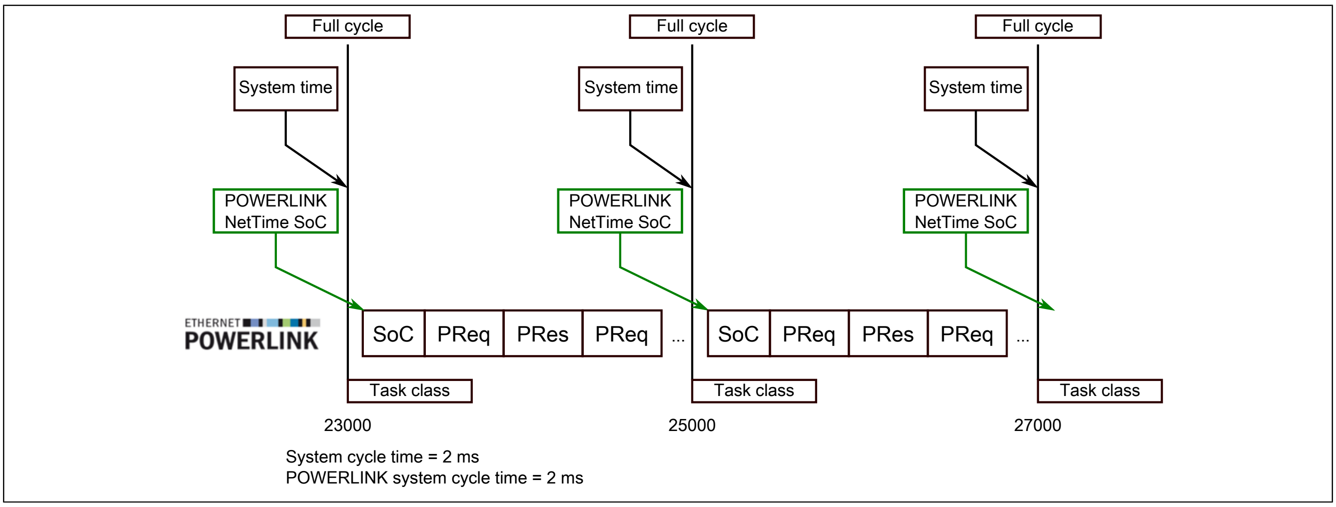
4.12.1.2 X2X Link - Reference moment



The reference moment on the X2X Link network is always calculated at the half cycle of the X2X Link cycle. This results in a difference between the system time and the X2X Link reference moment when the reference time is read out.

In the example above, this results in a difference of 1 ms, i.e. if the system time and X2X Link reference moment are compared at time 25000 in the task, then the system time returns the value 25000 and the X2X Link reference moment returns the value 24000.

4.12.1.3 POWERLINK reference moment

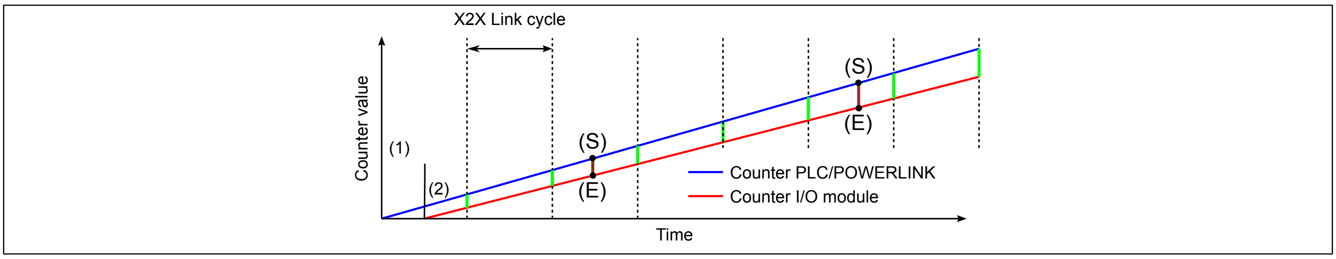


The reference moment on the POWERLINK network is always calculated at the start of cycle (SoC) of the POWERLINK network. The SoC starts 20 µs after the system tick. This results in the following difference between the system time and the POWERLINK reference time:

POWERLINK reference time = System time - POWERLINK cycle time + 20 µs.

In the example above, this means a difference of 1980 µs, i.e. if the system time and POWERLINK reference moment are compared at time 25000 in the task, then the system time returns the value 25000 and the POWERLINK reference moment returns the value 23020.

4.12.1.4 Synchronization of system time/POWERLINK time and I/O module



At startup, the internal counters for the PLC/POWERLINK (1) and the I/O module (2) start at different times and increase the values with microsecond resolution.

At the beginning of each X2X Link cycle, the PLC or the POWERLINK network sends time information to the I/O module. The I/O module compares this time information with the module's internal time and forms a difference (green line) between the two times and stores it.

When a NetTime event (E) occurs, the internal module time is read out and corrected with the stored difference value (brown line). This means that the exact system moment (S) of an event can always be determined, even if the counters are not absolutely synchronous.

Note

The deviation from the clock signal is strongly exaggerated in the picture as a red line.

4.12.2 Timestamp functions

NetTime-capable modules provide various timestamp functions depending on the scope of functions. If a timestamp event occurs, the module immediately saves the current NetTime. After the respective data is transferred to the CPU, including this precise moment, the CPU can then evaluate the data using its own NetTime (or system time), if necessary.

4.12.2.1 Time-based inputs

NetTime Technology can be used to determine the exact moment of a rising edge at an input. The rising and falling edges can also be detected and the duration between 2 events can be determined.

> **Information:**
>
> **The determined moment always lies in the past.**

4.12.2.2 Time-based outputs

NetTime Technology can be used to specify the exact moment of a rising edge on an output. The rising and falling edges can also be specified and a pulse pattern generated from them.

> **Information:**
>
> **The specified time must always be in the future, and the set X2X Link cycle time must be taken into account for the definition of the moment.**

4.12.2.3 Time-based measurements

NetTime Technology can be used to determine the exact moment of a measurement that has taken place. Both the starting and end moment of the measurement can be transmitted.

4.13 Minimum X2X cycle time

The minimum X2X cycle time is strongly dependent on the configured functions and the resulting load on the module. Setting "Fast reaction" and a very short system cycle (<50 μs) generally have a negative effect on the minimum X2X cycle time. This can result in error behavior with short X2X cycle times.