X20DS1319

1 General information

The module being used is a multifunctional digital signal processor module. It's flexibility allows it to be implemented for a wide range of tasks involving the creation or processing of digital signals. For example, two main uses include encoder emulation and controlling stepper output stages with pulse and direction signals. When used for encoder emulation, frequency inverters or servo axes with the speed follow function can follow a real or virtual master axis.

A further important feature is the timestamp function, which is integrated in the module. It can be used, for example, to create ramp curves for the counter in the encoder emulation virtually independent of bus cycle times. It's only necessary to enter the target counter value and the time at which it should be reached. The module generates the appropriate counter values, precisely in microsecond resolution and independently of the bus clock.

- 4 digital input channels
- 4 digital channels, configurable as inputs or outputs
- 1 universal counter pair (2 event counters, AB counter or up/down counter)
- Linear movement generator (A/B; direction/frequency) with up to 2 reference pulses
- SSI absolute encoder

2 Order data

Model number	Short description	Figure
	Digital signal processing and preparation	
X20DS1319	X20 multifunctional digital signal processor, 4 digital input channels, 4 digital channels configurable as inputs or outputs, 2 event counters, 1 universal counter pair configurable as A/B or up/down counter, linear movement generator (A/B, direction/frequency) with 2 reference pulses, SSI absolute encoder, NetTime module	
	Required accessories	
	Bus modules	
X20BM11	X20 bus module, 24 VDC keyed, internal I/O supply continuous	
X20BM15	X20 bus module, with node number switch, 24 VDC keyed, internal I/O supply continuous	
	Terminal blocks	
X20TB12	X20 terminal block, 12-pin, 24 VDC keyed	

Table 1: X20DS1319 - Order data

3 Technical data

Model number	X20DS1319
Short description	
I/O module	4 digital input channels, 4 digital channels configurable as inputs or outputs, 1 universal counter pair (2 event counters, AB counter or up/down counter), linear movement generator (A/B; direction/frequency) with up to two reference pulses, SSI absolute encoder, relative or absolute times of input edges in µs resolution, time-triggered I/O, I/O oversampling
General information	
B&R ID code	0x2547
Status indicators	I/O function per channel, operating state, module status
Diagnostics	
Module run/error	Yes, using status LED and software
Outputs	Yes, using status LED
Power consumption	
Bus	0.01 W
Internal I/O	1.5 W
Additional power dissipation caused by the actuators (resistive) [W]	-
Electrical isolation	
Channel - Bus	Yes
Channel - Channel	No
Type of signal lines	Shielded cables must be used for all signal lines.
Certification	
CE	Yes
KC	Yes
UL	cULus E115267 Industrial control equipment
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta = 0 - Max. 60°C FTZÚ 09 ATEX 0083X
DNV GL	Temperature: **B** (0 - 55°C) Humidity: **B** (up to 100%) Vibration: **B** (4 g) EMC: **B** (Bridge and open deck)
LR	ENV1
GOST-R	Yes
Linear movement generator	
Quantity	1
Encoder outputs	24 V, asymmetrical (A/B; direction/frequency)
Counter size	16/32-bit
Digital inputs	
Quantity	4 + 4, configurable as inputs or outputs using software
Nominal voltage	24 VDC
Input voltage	24 VDC -15 % / +20 %
Input current at 24 VDC	Approx. 1.3 mA
Input filter	
Hardware	≤2 µs
Software	-
Input circuit	Sink
Additional functions	SSI absolute encoder, universal counter pair, latch function for universal counter pair
Input resistance	18.4 kΩ
Input frequency	100 kHz
Switching threshold	
Low	<5 VDC
High	>15 VDC
Overload characteristics of encoder power supply	Short circuit protection, overload protection
Isolation voltage between channel and bus	500 V_{eff}
SSI absolute encoder	
Quantity	1
Counter size	Encoder-dependent up to 32-bit
Max. transfer rate	125 kbit/s
Encoder power supply	Module-internal, max. 600 mA
Nominal voltage	24 V, asymmetrical
Universal counter pair	
Quantity	1
Operating modes	2x event counter, up/down counter, AB counter
Encoder inputs	24 V, asymmetrical
Counter size	16/32-bit
Input frequency	Max. 100 kHz

Table 2: X20DS1319 - Technical data

Model number	X20DS1319
Evaluation	
AB counter	4x
Event counter	2x
Up/Down counter	2x
Signal form	Square wave pulse
Encoder power supply	Module-internal, max. 600 mA
Digital outputs	
Design	Push / Pull / Push-Pull
Quantity	Up to 4, configurable as inputs or outputs using software
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 cutoff if overcurrent or short circuit occurs, integrated protection for switching inductances
Diagnostic status	Output monitoring
Leakage current when switched off	Max. 25 µA
Residual voltage	<0.9 V at 0.1 A rated current
Peak short circuit current	<10 A
Switching on after overload or short circuit cutoff	Approx. 10 ms (depends on the module temperature)
Switching delay	
0 -> 1	<2 µs
1 -> 0	<2 µs
Switching frequency	
Resistive load	Max. 125 kHz
Inductive load	See section "Switching inductive loads"
Braking voltage when switching off inductive loads	Switching voltage + 0.6 VDC
Additional functions	Timing for SSI absolute encoder, linear movement generator
Operating conditions	
Mounting orientation	
Horizontal	Yes
Vertical	Yes
Installation at elevations above sea level	
0 to 2000 m	No limitations
>2000 m	Reduction of ambient temperature by 0.5°C per 100 m
EN 60529 protection	IP20
Environmental conditions	
Temperature	
Operation	
Horizontal installation	-25 to 60°C
Vertical installation	-25 to 50°C
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 characteristics	
Note	Order 1x X20TB12 terminal block separately Order 1x X20BM11 bus module separately
Spacing	$12.5^{+0.2}$ mm

Table 2: X20DS1319 - Technical data

4 LED status indicators

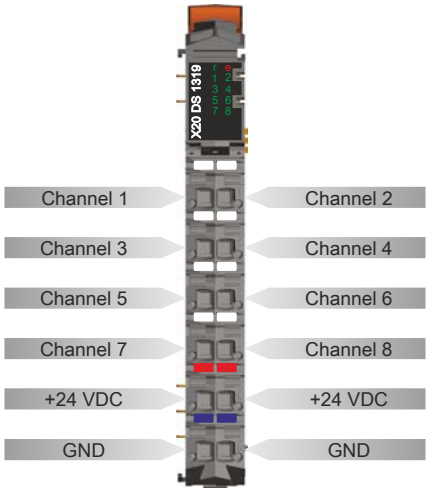
For a description of the various operating modes, see section "Additional information - Diagnostic LEDs" of the X20 system user's manual.

Figure	LED	Color	Status	Description
	r	Green	Off	No power to module
			Single flash	RESET mode
			Double flash	BOOT mode (during firmware update)[1]
			Blinking	PREOPERATIONAL mode
			On	RUN mode
	e	Red	Off	No power to module or everything OK
			Single flash	I/O error. Possible causes: • SSI error[2]
			Double flash	System error. Possible causes: • Motion function error[3] • I/O oversampling error[4] • Edge detection error[4]
			Triple flash	I/O error and system error occur together
			On	Error or reset status
	1 - 8	Green		Status of the corresponding digital signal

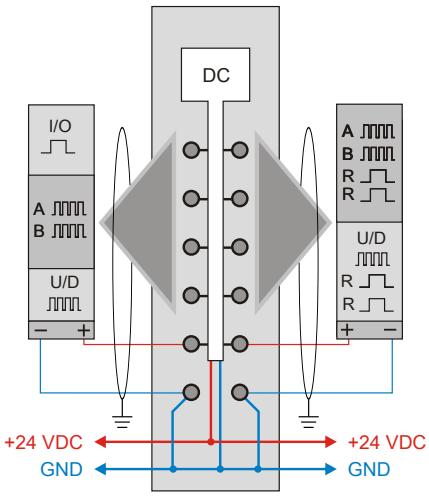
1) Depending on the configuration, a firmware update can take up to several minutes.
2) See "Error state - SSI" on page 13 register for the exact error description.
3) See "Error state - Motion functions" on page 13 register for the exact error description.
4) See "Error state - Output data and edge detection" on page 12 register for the exact error description.

5 Pinout

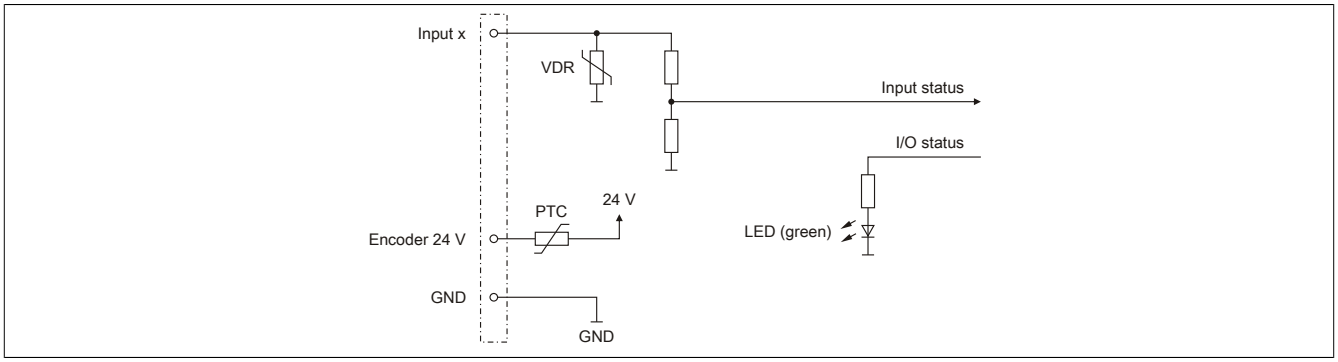
Shielded cables must be used for all signal lines.



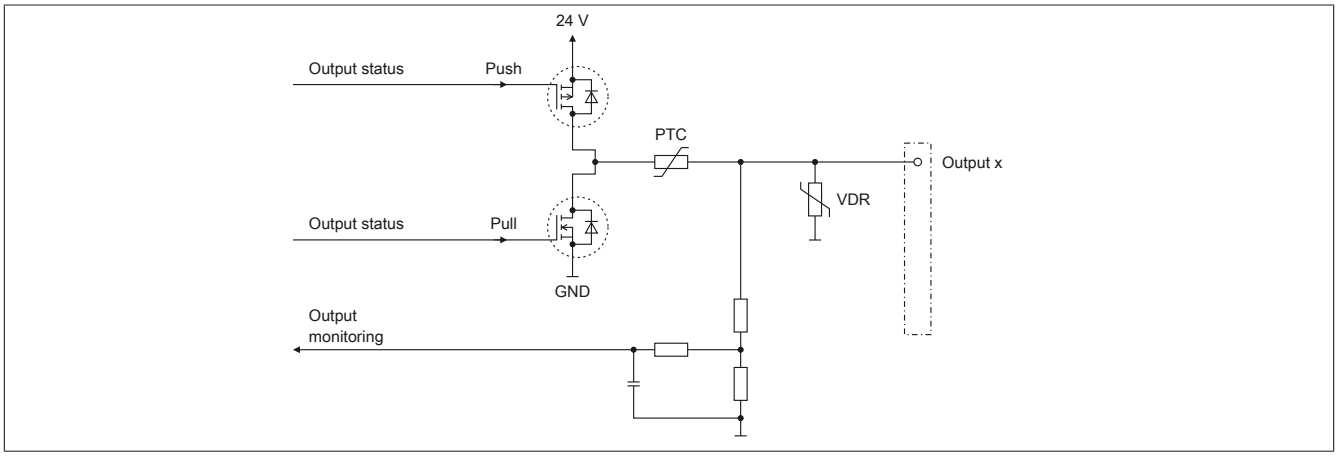
6 Connection example



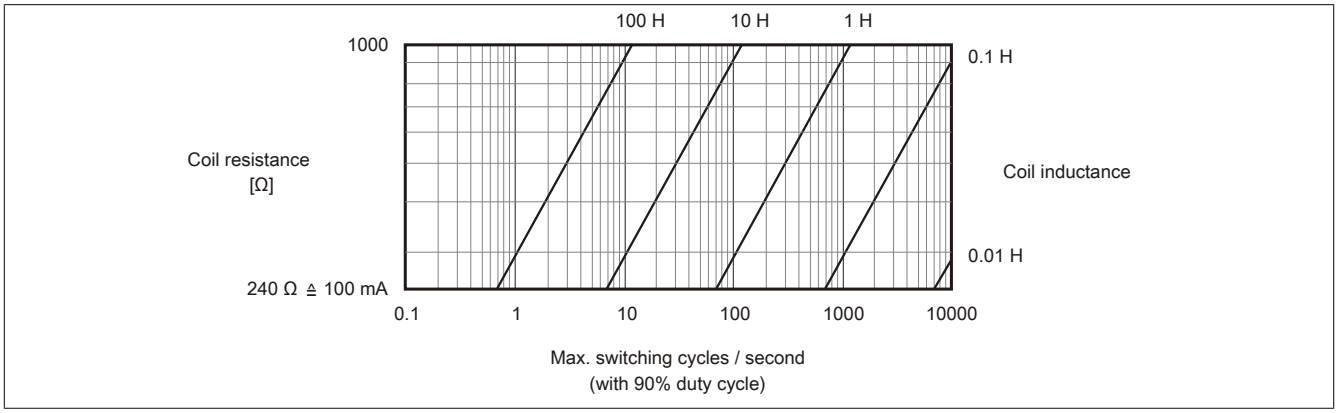
7 Input circuit diagram



8 Output circuit diagram



9 Switching inductive loads



10 Connection options

Digital input/output

Channel	Function
1	Input
2	Input
3	Input / Output
4	Input / Output
5	Input
6	Input
7	Input / Output
8	Input / Output

Wiring of the SSI absolute encoder

Channel	Function
5 (input)	Data
7 (output)	Clock

Wiring of the linear movement generator

Channel	Up-Down	AB
3 (output)	Direction	A
4 (output)	Frequency	B
7 (output)	Reference 1	
8 (output)	Reference 2	

Wiring of the universal counter pair

Channel	Edge counters	Up/Down counter	Incremental
1 (input)	Input 1	Direction	A
2 (input)	Input 2	Frequency	B
5 (input)		Latch input 1 (R)	
6 (input)		Latch input 2 (E)	

11 Register description

11.1 General data points

In addition to the registers listed in the register description, the module also has other more general data points. These registers are not specific to the module but contain general information such as serial number and hardware version.

These general data points are listed in section "Additional information - General data points" of the X20 system user's manual.

11.2 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						
1537	CfO_EdgeDetectPollCycleID	USINT				•
1548	CfO_EdgeDetectEventEnable	UDINT				•
1665 + (N-1) * 16	CfO_EdgeDetectUnit0NMode (Index N = 1 to 4)	USINT				•
1667 + (N-1) * 16	CfO_EdgeDetectUnit0NLeading (Index N = 1 to 4)	USINT				•
1669 + (N-1) * 16	CfO_EdgeDetectUnit0NMaster (Index N = 1 to 4)	USINT				•
1671 + (N-1) * 16	CfO_EdgeDetectUnit0NSlave (Index N = 1 to 4)	USINT				•
Configuration - Movement functions						
4097	CfO_FifoSize	USINT				•
4099	CfO_Mode	SINT				•
4101	CfO_SpeedLimit	USINT				•
4103	CfO_FormatAdjust	USINT				•
4105	CfO_TimeStampRange	SINT				•
4107	CfO_PositionRange	SINT				•
4109	CfO_Reference0Range	SINT				•
4111	CfO_Reference1Range	SINT				•
4116	CfO_TimeStampDelay	DINT				•
4124	CfO_SpeedCycleTime_32bit	UDINT				•
4129	CfO_ResolPosition	SINT				•
4131	CfO_ResolSpeed	SINT				•
4220	CfO_AccelDataInit	UDINT				•
4260	CfO_Reference0Start	DINT				•
4268	CfO_Reference0StopMargin	DINT				•
4276	CfO_Reference1Start	DINT				•
4284	CfO_Reference1StopMargin	DINT				•
Configuration - SSI						
2049	CfO_CycleSelect	USINT				•
2051	CfO_PhysicalMode	USINT				•
2053	CfO_DataBits	USINT				•
2055	CfO_NullBits	USINT				•
Configuration - Universal counter						

Register	Name	Data type	Read		Write	
			Cyclic	Acyclic	Cyclic	Acyclic
6145	CfO_CounterCycleSelect	USINT				•
6147	CfO_CounterMode	USINT				•
6149	CfO_LatchMode	USINT				•
6151	CfO_LatchComparator	USINT				•
6153	CounterControl	USINT			•	
	CounterReset	Bit 0				
	LatchEnable	Bit 1				
Communication - General						
546	ProtocolError (16-bit)	USINT	•			
547	ProtocolError (8-bit)	UINT	•			
550	ProtocolSequenceViolation (16-bit)	UINT	•			
551	ProtocolSequenceViolation (8-bit)	USINT	•			
Communication - Error register						
257	Error state - Output data and edge detection	USINT	•			
	OutputControlError	Bit 4				
	OutputCopyError	Bit 5				
	EdgeDetectError	Bit 6				
259	Error state - SSI	USINT	•			
	SSICycleTimeViolation	Bit 0				
	SSIParityError	Bit 1				
261	Error state - Movement functions	USINT	•			
	MovFifoEmpty	Bit 0				
	MovFifoFull	Bit 1				
	MovTargetTimeViolation	Bit 2				
	MovMaxFrequencyViolation	Bit 3				
321	Acknowledge error messages - Output data and edge detection	USINT			•	
	QuitOutputControlError	Bit 4				
	QuitOutputCopyError	Bit 5				
	QuitEdgeDetectError	Bit 6				
323	Acknowledge error messages - SSI	USINT			•	
	SSIQuitCycleTimeViolation	Bit 0				
	SSIQuitParityError	Bit 1				
325	Acknowledge error messages - Movement functions	USINT			•	
	MovQuitFifoEmpty	Bit 0				
	MovQuitFifoFull	Bit 1				
	MovQuitTargetTimeViolation	Bit 2				
	MovQuitMaxFrequencyViolation	Bit 3				
Communication - System timer						
683	SDCLifeCount	SINT	•			
Communication - Direct I/O						
915	"DigitalOutput" register	USINT			•	
	DigitalOutput03	Bit 2				
	DigitalOutput04	Bit 3				
	DigitalOutput07	Bit 6				
	DigitalOutput08	Bit 7				
927	"DigitalInput" register	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	•			

Register	Name	Data type	Read		Write	
			Cyclic	Acyclic	Cyclic	Acyclic
1148 + N	OversampleInput0NSample8_1 (Index N = 1 to 4)	USINT	•			
Communication - Edge detection						
1794 + (N-1) * 32	EdgeDetect0NMastercount (16-bit) (Index N = 1 to 4)	INT	•			
1795 + (N-1) * 32	EdgeDetect0NMastercount (8-bit) (Index N = 1 to 4)	SINT	•			
1798 + (N-1) * 32	EdgeDetect0NSlavecount (16-bit) (Index N = 1 to 4)	INT	•			
1799 + (N-1) * 32	EdgeDetect0NSlavecount (8-bit) (Index N = 1 to 4)	SINT	•			
1804 + (N-1) * 32	EdgeDetect0NDifference (32-bit) (Index N = 1 to 4)	DINT	•			
1806 + (N-1) * 32	EdgeDetect0NDifference (16-bit) (Index N = 1 to 4)	INT	•			
1812 + (N-1) * 32	EdgeDetect0NMastertime (32-bit) (Index N = 1 to 4)	DINT	•			
1814 + (N-1) * 32	EdgeDetect0NMastertime (16-bit) (Index N = 1 to 4)	INT	•			
1820 + (N-1) * 32	EdgeDetect0NSlavetime (32-bit) (Index N = 1 to 4)	DINT	•			
1822 + (N-1) * 32	EdgeDetect0NSlavetime (16-bit) (Index N = 1 to 4)	INT	•			
Communication - Movement functions						
4225	MovementControl	USINT			•	
	MovPosEnable	Bit 0				
	MovSpeedEnable	Bit 1				
4244	MovTargetTime (32-bit)	DINT			•	
4246	MovTargetTime (16-bit)	INT			•	
4252	MovTargetPosition (32-bit)	DINT			•	
4254	MovTargetPosition (16-bit)	INT			•	
4260	MovReference1Start (32-bit)	DINT			•	
4262	MovReference1Start (16-bit)	INT			•	
4268	MovReference1StopMargin (32-bit)	DINT			•	
4270	MovReference1StopMargin (16-bit)	INT			•	
4276	MovReference2Start (32-bit)	DINT			•	
4278	MovReference2Start (16-bit)	INT			•	
4284	MovReference2StopMargin (32-bit)	DINT			•	
4286	MovReference2StopMargin (16-bit)	INT			•	
4212	MovSpeed (32-bit)	DINT			•	
4210	MovSpeed (16-bit)	INT			•	
4220	MovAcceleration (32-bit)	UDINT			•	
4218	MovAcceleration (16-bit)	UINT			•	
4292	MovTimeValid (32-bit)	DINT	•			
4294	MovTimeValid (16-bit)	INT	•			
4300	MovPosition (32-bit)	DINT	•			
4302	MovPosition (16-bit)	INT	•			
Communication - SSI						
2084	SSITimeValid (32-bit)	DINT	•			
2086	SSITimeValid (16-bit)	INT	•			
2092	SSITimeChanged (32-bit)	DINT	•			
2094	SSITimeChanged (16-bit)	INT	•			
2100	SSIPosition (32-bit)	(U)DINT	•			
2102	SSIPosition (16-bit)	UINT	•			
Communication - Universal counter						
6303	LatchCount	SINT	•			
6308	CounterTimeValid (32-bit)	DINT	•			
6310	CounterTimeValid (16-bit)	INT	•			
6324	Counter01TimeChanged (32-bit)	DINT	•			
6326	Counter01TimeChanged (16-bit)	INT	•			
6332	Counter02TimeChanged (32-bit)	DINT	•			
6334	Counter02TimeChanged (16-bit)	INT	•			
6340	CounterValue01 (32-bit)	DINT	•			
6342	CounterValue01 (16-bit)	INT	•			
6348	CounterValue02 (32-bit)	DINT	•			
6350	CounterValue02 (16-bit)	INT	•			
6356	CounterLatch01 (32-bit)	DINT	•			
6358	CounterLatch01 (16-bit)	INT	•			
6364	CounterLatch02 (32-bit)	DINT	•			
6366	CounterLatch02 (16-bit)	INT	•			
6372	CounterRel01 (32-bit)	DINT	•			
6374	CounterRel01 (16-bit)	INT	•			
6380	CounterRel02 (32-bit)	DINT	•			
6382	CounterRel02 (16-bit)	INT	•			

11.3 General

11.3.1 Use with Automation Studio

The module is supported via X2X Link and POWERLINK.

X2X Link supports a up to 28 bytes of synchronous data per module. To optimize use and to prevent needless data transfer, the data points can be adjusted as needed in Automation Studio. Data points that are not needed can be disabled, and the bit width of the data points can be defined.

11.3.2 Timestamp function

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

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 time defined by the CPU.

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

11.3.3 Synchronization jitter

Because the CPU – which determines the X2X net time – and the module have different clocks, the module's internal X2X net time must be synchronized with the CPU's net time. Due to this synchronization, the module's internal X2X net time is corrected by a maximum of 1/8 µs per system cycle if necessary. This synchronization jitter becomes noticeable when using the net time 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 the "CfO_EdgeDetectUnitMode" on page 27 register).

11.4 General registers

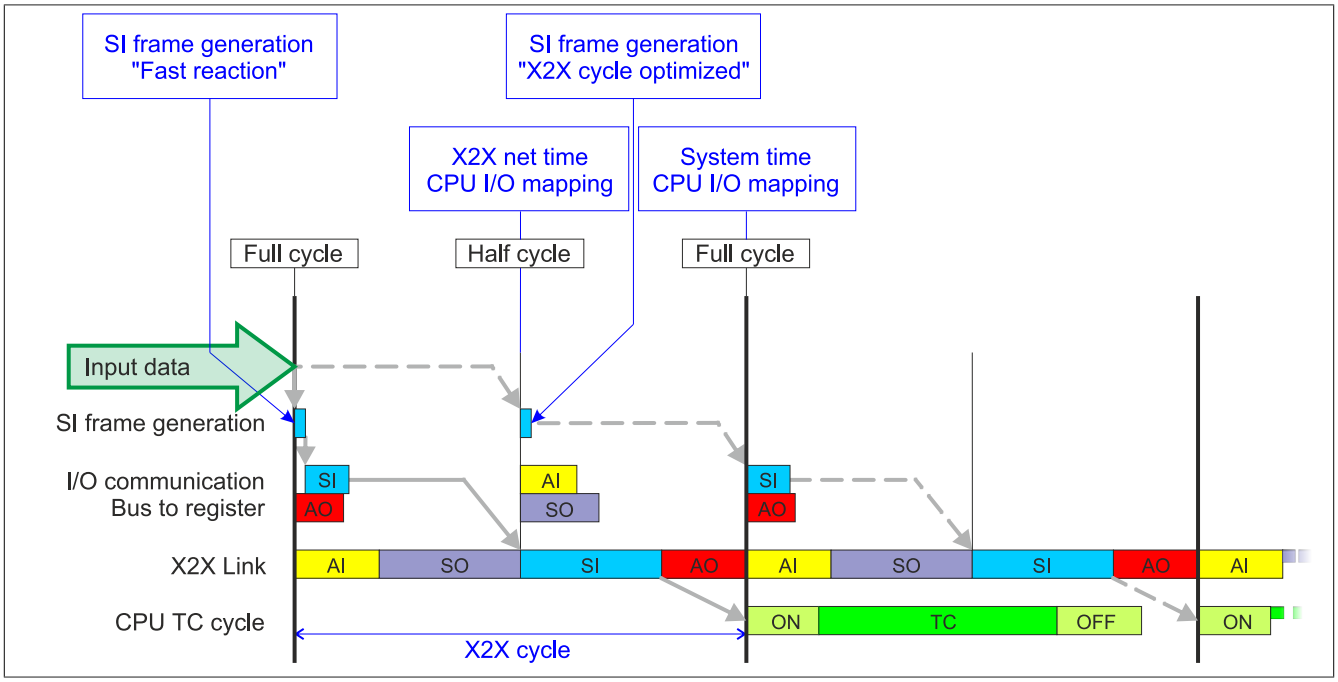
11.4.1 Defining the moment for generating synchronous input data

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

This register determines when the synchronous input data is generated for transfer. This has a decisive effect on the timing of the input data.

The 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	Value	Information
USINT	9	X2X cycle optimized
	14	Fast reaction



11.4.2 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, the "Network information" parameter 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	Value	Information
USINT	0 to 255	Error counter (8-bit)
UINT	0 to 65535	Error counter (16-bit)

11.4.3 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, the "Network information" parameter 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	Value	Information
USINT	0 to 255	Error counter (8-bit)
UINT	0 to 65535	Error counter (16-bit)

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

Name:
SDCLifeCount

Counter that is incremented with each system timer cycle. The "SDC information" setting in the Automation Studio I/O configuration can be used to activate this register in the I/O mapping as the 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	Value
SINT	-128 to 127

11.5 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 accordingly 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 source of the error persists, then the error bit is set again as soon as the error is detected again (i.e. cannot be reset).

Acknowledging the error does not affect the module's functionality. If possible, the module automatically resumes processing as soon as the source of the error has been corrected.

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 source of the error has been corrected.

11.5.1 Error state register - Output data and edge detection

Name:
OutputControlError
OutputCopyError
EdgeDetectError

Errors in the output data and cycle time settings are indicated in this register.

Data type	Values
USINT	See 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 while in the mode "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. (e.g. an attempt was made to write oversampling output data to an address outside of the OversampleOutputWindow).
6	EdgeDetectError	0	No error
		1	Cycle time violation edge detection: The "EdgeDetectPollCycle" must be ≤255 µs. This error is caused if the cycle defined in the "CfO_EdgeDetectPollCycleID" on page 26 register is >255 µs.
7	Reserved	-	

11.5.2 Error state register - SSI

Name:
SSICycleTimeViolation
SSIParityError

SSI interface errors are indicated in this register.

Data type	Values
USINT	See bit structure.

Bit structure:

Bit	Description	Value	Information
0	SSICycleTimeViolation	0	No error
		1	Error occurred, possible causes: • SSI transfer takes longer than the defined "Update cycle". • Monoflop check is enabled and the SSI data line does not assume the defined level after the transfer is complete.
1	SSIParityError	0	No error
		1	SSI parity error
2 - 7	Reserved	-	

11.5.3 Error state register - Movement functions

Name:
MovFifoEmpty
MovFifoFull
MovTargetTimeViolation
MovMaxFrequencyViolation

Movement function errors are indicated in this register.

Data type	Values
USINT	See bit structure.

Bit structure:

Bit	Description	Value	Information
0	MovFifoEmpty	0	No error
		1	The position/timestamp FIFO is empty.
1	MovFifoFull	0	No error
		1	The position/timestamp FIFO has exceeded the size defined in the "FifoSize" on page 31 register.
2	MovTargetTimeViolation	0	No error
		1	This only occurs when the "MovTargetTime" on page 36 is in the past.
3	MovMaxFrequencyViolation	0	No error
		1	The maximum output frequency setpoint has exceeded the maximum frequency configured in the "CfO_SpeedLimit" on page 32 register.
4 - 7	Reserved	-	

11.5.4 Acknowledge error message register - Output data and edge detection

Name:
QuitOutputControlError
QuitOutputCopyError
QuitEdgeDetectError

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

Data type	Values
USINT	See 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	-	

11.5.5 Acknowledge error message register - SSI

Name:
SSIQuitCycleTimeViolation
SSIQuitParityError

Error messages from the "Error state - SSI" on page 13 register can be acknowledged by setting the corresponding bits in this register.

Data type	Values
USINT	See bit structure.

Bit structure:

Bit	Description	Value	Information
0	SSIQuitCycleTimeViolation	0	No change
		1	Acknowledge error
1	SSIQuitParityError	0	No change
		1	Acknowledge error
2 - 7	Reserved	-	

11.5.6 Acknowledge error message register - Movement functions

Name:
MovQuitFifoEmpty
MovQuitFifoFull
MovQuitTargetTimeViolation
MovQuitMaxFrequencyViolation

Error messages from the "Error state register - Movement functions" on page 13 register can be acknowledged by setting the corresponding bits in this register.

Data type	Values
USINT	See bit structure.

Bit structure:

Bit	Description	Value	Information
0	MovQuitFifoEmpty	0	No change
		1	Acknowledge error
1	MovQuitFifoFull	0	No change
		1	Acknowledge error
2	MovQuitTargetTimeViolation	0	No change
		1	Acknowledge error
3	MovQuitMaxFrequencyViolation	0	No change
		1	Acknowledge error
4 - 7	Reserved	-	

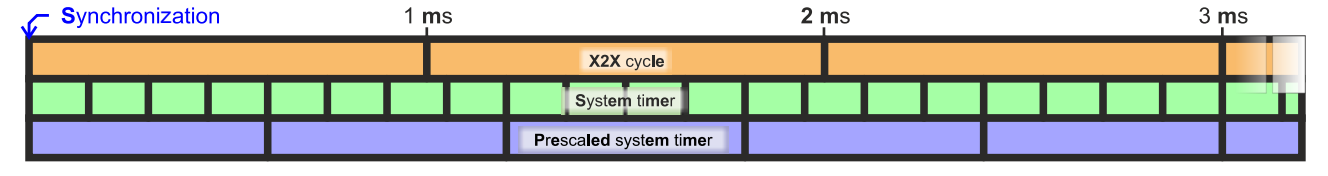
11.6 System timer

The module's individual functions all depend on a system timer. This internal "system cycle time" can be defined from 25 to 255 µs. The functions can also be run with the help of 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 X2X net time 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.

The following values apply to the following example:

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



11.6.1 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 this register in steps of 1/8 µs. 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	Value	Information
UINT	200 to 2047	System timer cycle time in steps of 1/8 µs (25 to 255.875 µs)

11.6.2 Offsetting the synchronization moment of the system cycle

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

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

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

11.6.3 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 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	Value	Information
UINT	2 to 128	Multiple of the system timer

11.7 Physical I/O configuration

11.7.1 "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 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 to 15	See: Overview of output channel functions

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

Overview of output channel functions

Values of bits 4 to 7	Output channel 3	Output channel 4	Output channel 7	Output channel 8
0	Direct I/O			
1				SSI clock output
2	ABR emulation (A)	ABR emulation (B)	ABR emulation (reference 1)	ABR emulation (reference 2)
3	Up/down emulation (direction)	Up/down emulation (frequency)	Up/down emulation (reference 1)	Up/down emulation (reference 2)
4 - 15	Reserved			

11.8 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).

11.8.1 "EdgeGenTimestamp" register

Name:
CfO_DirectIOClearMask0_7
"Direct control of output channel 03" to "Direct control 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 is reset ("output control channel 7_0" on page 17 or "DigitalOutput0x" register in the Automation Studio I/O mapping).

Data type	Values
USINT	See 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

11.8.2 "CfO_DirectIOSetMask0_7" register

Name:
CfO_DirectIOSetMask0_7
"Direct control of output channel 03" to "Direct control 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 is set ("output control channel 7_0" on page 17 or "DigitalOutput0x" register in the Automation Studio I/O mapping).

Data type	Values
USINT	See 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

11.8.3 "DigitalOutput" register

Name:
DigitalOutput03 and DigitalOutput04, DigitalOutput07 and DigitalOutput08

This register contains the bits for controlling the direct I/O output channels. Depending on how the "CfO_DirectIOClearMask0_7" on page 16 and "CfO_DirectIOSetMask0_7" on page 17 registers are configured, the digital outputs are set to the status of the respective bits in this register.

Data type	Values
USINT	See bit structure.

Bit structure:

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

11.8.4 "DigitalInput" register

Name:
DigitalInput01 to DigitalInput08

This register displays the status of the digital input channels.

Data type	Values
USINT	See bit structure.

Bit structure:

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

11.9 Oversampled I/O

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

In "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.

11.9.1 Addressing the output control buffer

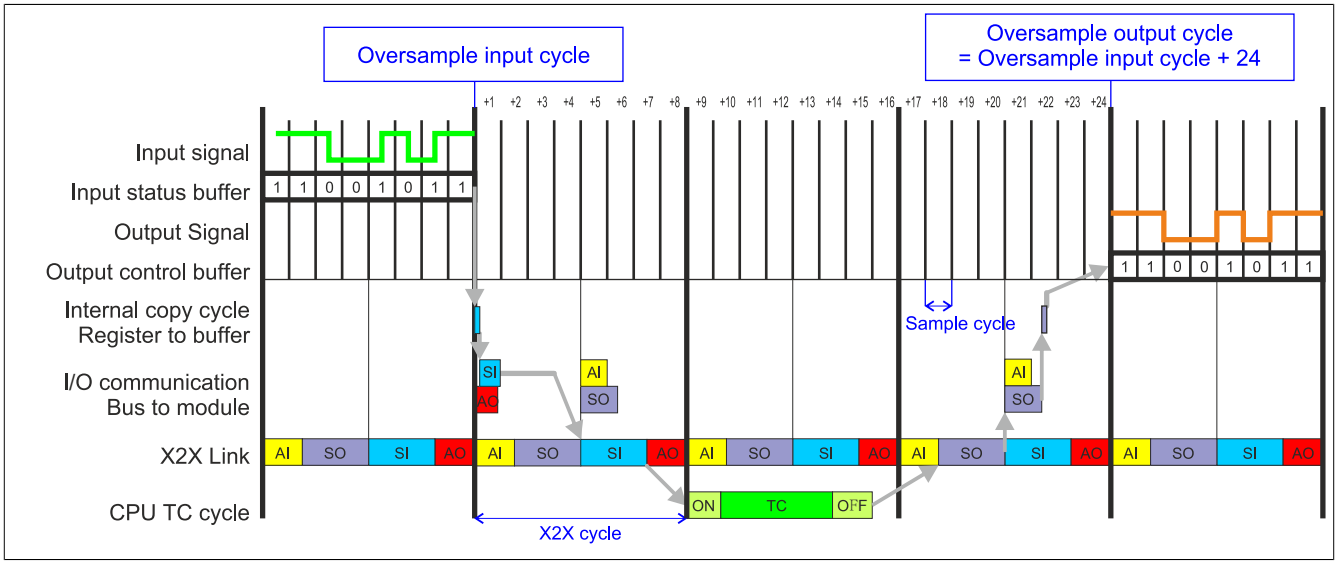
The module has one cyclic 256-bit output control buffer for each oversample channel. One bit is output from these buffers to the configured physical output channels in each "sample cycle". When new data is transferred to one of these buffers, the application must define where in the respective buffer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

11.9.1.1 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 25 registers) an address must also be transferred in the "OversampleOutputCycle" on page 24 register. This address determines where in the output control buffer the new data should be copied to. In order to calculate this address, you must account for the contents of the "OversampleInputCycle" on page 25 register, which contains the address of the most recently output data, and the transfer time to the module. To help avoid incorrect addressing of the output control buffer, the buffer section that is capable of being written to can be limited using the "OversampleOutputWindow" on page 22 register. 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.

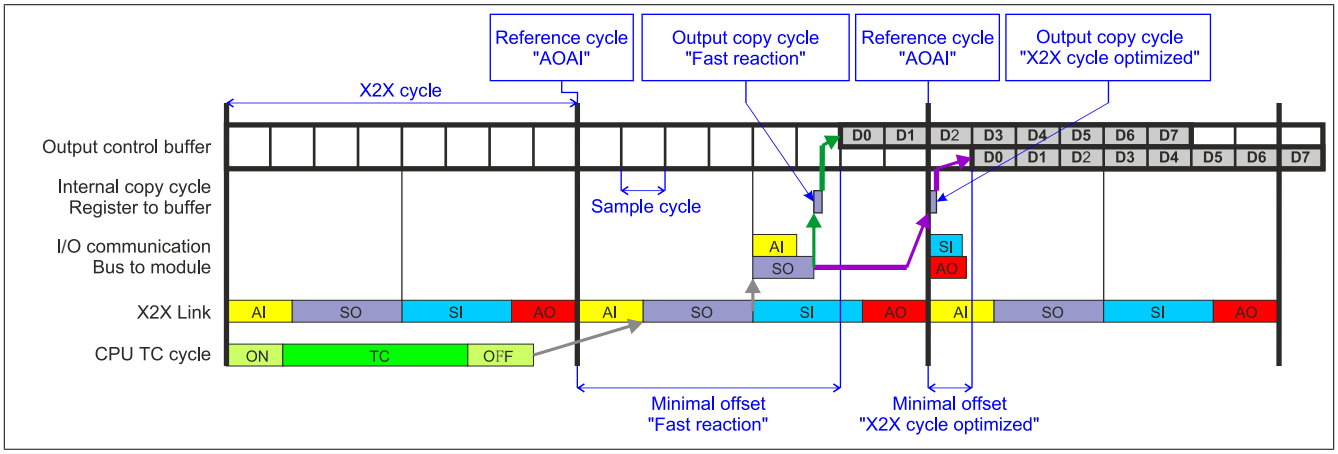
Example

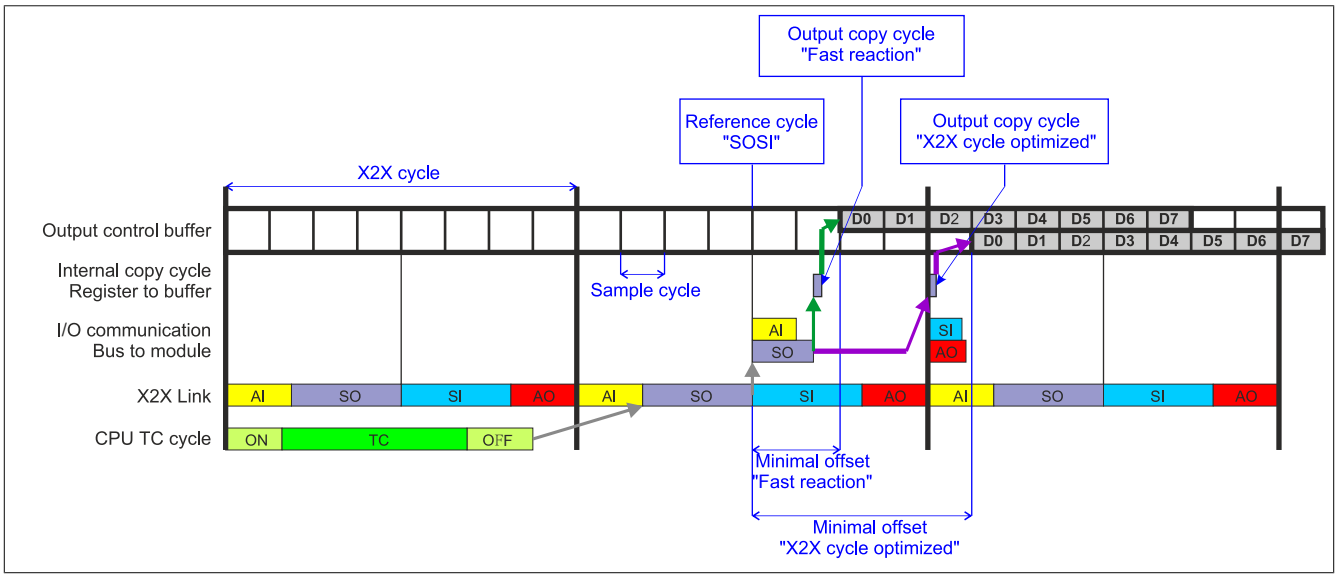
Timing from oversample input cycle to oversample output cycle in absolute output mode ("SI-frame generation = Fast reaction", "Output copy cycle = Fast reaction", 8 samples per X2X cycle):



11.9.1.2 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 defined output copy cycle time. The "OversampleSample-Offset" on page 24 register serves as the offset. The new data cannot start being output immediately at the output copy cycle time 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.





11.9.2 "CfO_OversampleMode" register

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 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 such a situation, the output assumes the "Output default state" configured in the "CfO_OversampleConfigOutput" on page 23 register.

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.

11.9.3 "CfO_OversampleSampleCycleID" register

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

The source of the sample cycle can be 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 entered in one bit of the respective input status buffer.

Data type	Value	Information
USINT	2	System timer The value configured in the "CfO_SystemCycleTime" on page 15 register 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.

11.9.4 "CfO_OversampleRelativeCycleID" register

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

The source of the user interface reference cycle can be configured in this register.

- The input data is referenced at the time of the "reference cycle". The referenced data is then copied to the "oversample input sample register" on page 26 at the time of SI frame generation, while taking the OversampleInputWindow into account.
- 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 with it also the output data production and input data acquisition (e.g. to the X2X cycle).

Data type	Value	Information
USINT	2	System timer The value configured in the "CfO_SystemCycleTime" on page 15 register 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.

11.9.5 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 25 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	Value	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.

11.9.6 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 25 registers to the output control buffers at the time of the output copy cycle.

Data type	Value	Information
USINT	1 to 64	Output bits

11.9.7 "CfO_OversampleInputBits" register

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 26 register during SI frame generation.

Data type	Value	Information
USINT	1 to 64	Input bits

11.9.8 "CfO_OversampleOutputWindow" register

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

Determines the area in the output control buffer in which data can 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, with the setting "Output control mode = Single", this register is set to 128 bits and with the setting "Output control mode = Continuous" it is set to 255 bits.

Data type	Value	Information
USINT	0 to 255	Output window

11.9.9 "CfO_OversampleInputWindow" register

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

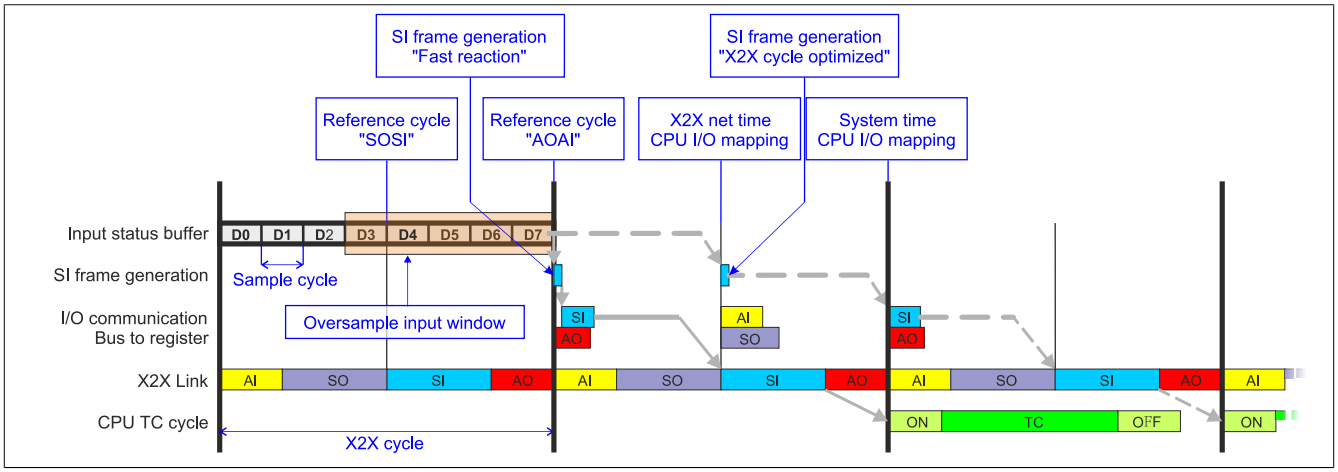
The "OversampleInputWindow" determines when the input data is referenced. It is located chronologically before SI frame generation. If the reference time ("reference cycle" on page 21) is within this window, then the referenced data is copied from the input status buffer to the "OversampleInput0NSample" on page 26 register. If the time at which the reference occurs is outside the "OversampleInputWindow" then the data that is most recent at the time of "SI frame generation" is copied from the input status buffer to the "OversampleInput0NSample" on page 26 register.
This register is limited internally with to the value set in the "CfO_OversampleInputBits" on page 22 register.

Information:

As a result, the OversampleInputTime and the OversampleInputCycle are set either at the reference time or at the time of "SI frame generation".

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

Data type	Value	Information
USINT	0 to 63	Input window



11.9.10 "CfO_OversampleConfigInput" register

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

This register determines which physical input channel an oversample I/O input should be linked to.

Data type	Values
USINT	See 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	-	

11.9.11 "CfO_OversampleConfigOutput" register

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 state" to "Oversample I/O 04 → Output default state" in the Automation Studio I/O configuration

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

The "Output default state" bits determine which level the respective output assumes before oversampling is started. Furthermore, the output is set to the defined "Output default state" in the event of an error.

Data type	Values
USINT	See 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	Output default state: Clear "Oversample I/O 0x →Output default state"	0	Output not cleared by default
		1	Output cleared by default
7	Output default state: Set "Oversample I/O 0x →Output default state"	0	Output not set by default
		1	Output set by default

11.9.12 Oversampling configuration

Name:
OversampleEnable
OversampleOutputValidate

This register can be used to configure oversampling and the copy procedure for the output buffer.

Data type	Values
USINT	See 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 25 registers in each X2X cycle.
2 - 7	Reserved	-	

11.9.13 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	Value	Information
USINT	0 to 255	Address of the output control buffer

11.9.14 "OversampleSampleOffset" register

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	Value	Information
USINT	0 to 255	Offset of output sample data

11.9.15 Oversample output sample data

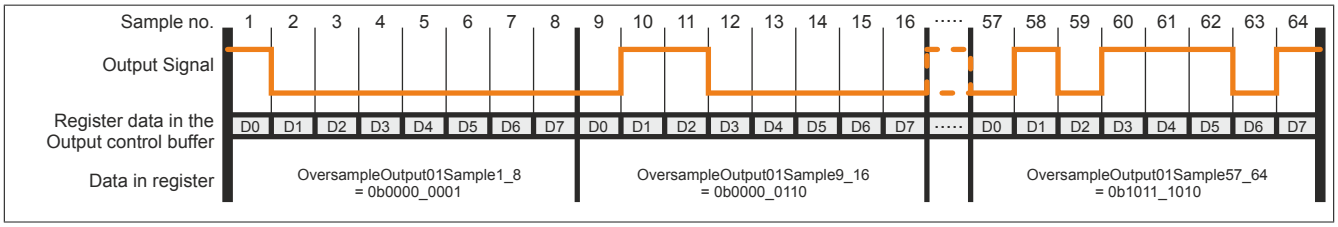
Name:
OversampleOutput01Sample1_8 to OversampleOutput04Sample1_8
OversampleOutput01Sample9_16 to OversampleOutput04Sample9_16
OversampleOutput01Sample17_24 to OversampleOutput04Sample17_24
OversampleOutput01Sample25_32 to OversampleOutput04Sample25_32
OversampleOutput01Sample33_40 to OversampleOutput04Sample33_40
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 during the 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 "OversampleOutputSample8_1" is copied to the output control buffer first, meaning that it is the first bit that is output. "OversampleOutputSample64_57" bit 7 is the last bit to be output.

Data type	Value	Information
USINT	0 to 255	Output sample data

Example

Assignment of "OversampleOutputSample" register data to output signal



11.9.16 "OversampleInputTime" register

Name:
OversampleInputTime

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

Data type	Value	Information
INT	-32,768 to 32,767	X2X net time of the input data

11.9.17 "OversampleInputCycle" register

Name:
OversampleInputCycle

This register provides the width of the input status buffer address for the input sample data.
Furthermore, the value in this register can be used for referencing an absolute addressing of the output control buffer.

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

11.9.18 "OversampleInputSample" register

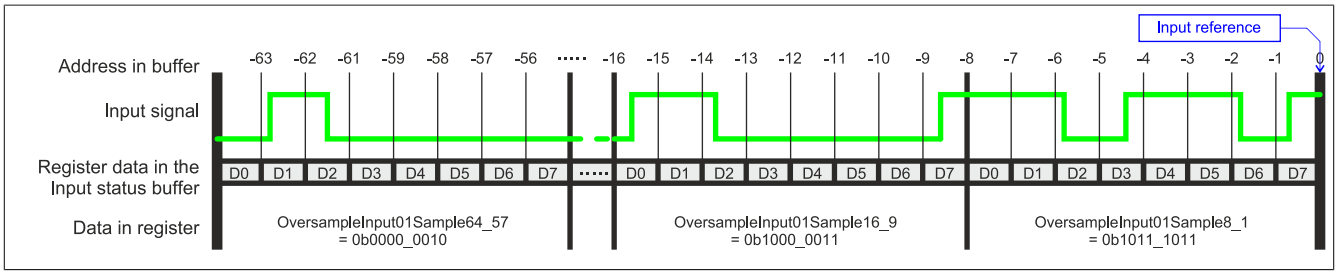
Name:
OversampleInput01Sample8_1 to OversampleInput04Sample8_1
OversampleInput01Sample16_9 to OversampleInput04Sample16_9
OversampleInput01Sample24_17 to OversampleInput04Sample24_17
OversampleInput01Sample32_25 to OversampleInput04Sample32_25
OversampleInput01Sample40_33 to OversampleInput04Sample40_33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

The data of the four oversample input status buffers are copied to this register at the time 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	Value	Information
USINT	0 to 255	Input sample data

Example

Input signal and resulting data in "OversampleInputSample"



11.10 Edge detection

The module's edge detection function identifies edges with µs precision. The concept is based on a maximum of 4 units. A master and a slave edge can be configured for each unit.

At each master edge, the net time of the master edge and the net time 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.

11.10.1 "CfO_EdgeDetectPollCycleID" register

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

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

> **Information:**
>
> **The polling cycle must be less than or equal to 255 µs. Setting the cycle > 255 µs causes an EdgeDetectError.**

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

11.10.2 "CfO_EdgeDetectEventEnable" register

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

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

In "event triggered" mode, the net time of each edge is recorded immediately an interrupt. 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 "polling" mode, only the net time 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 IO 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 bit structure.

Bit structure:

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

11.10.3 "CfO_EdgeDetectUnitMode" register

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 net time.

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

Data type	Values
USINT	See bit structure.

Bit structure:

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

11.10.4 "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 net time is always saved within the module. A FIFO is provided inside the module which always stores the last 256 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 255	Position in the slave edge FIFO

11.10.5 "CfO_EdgeDetectUnitMaster" register

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

This register is used to select the source of the master edge for the respective "edge detection unit".

Data type	Value	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

11.10.6 "CfO_EdgeDetectUnitSlave" register

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

This register is used to select the source of the slave edge for the respective "edge detection unit".

Data type	Value	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

11.10.7 "EdgeDetectMastercount" register

Name:
EdgeDetect01Mastercount to EdgeDetect04Mastercount

The reference pulses of the detected master edges are counted in this register.

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

11.10.8 "EdgeDetectSlavecount" register

Name:
EdgeDetect01Slavecount to EdgeDetect04Slavecount

Counts the number of detected slave edges consecutively. The contents of this register are only updated when a master edge occurs. 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)

11.10.9 "EdgeDetectDifference" register

Name:
EdgeDetect01Difference to EdgeDetect04Difference

Contains the time difference between a master edge and the last slave edge addressed via "Slave leading" on page 28.

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)

11.10.10 "EdgeDetectMastertime" register

Name:
EdgeDetect01Mastertime to EdgeDetect04Mastertime

The exact net time is copied in this register when a master edge occurs.

Data type	Value	Information
INT	-32,768 to 32,767	Net time of master edge (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Net time of master edge (32-bit)

11.10.11 "EdgeDetectSlavetime" register

Name:
EdgeDetect01Slavetime to EdgeDetect04Slavetime

When a master edge occurs, the exact net time of any slave edge that may have occurred prior to the master edge and addressed by "Slave leading" on page 28 is copied in this register. If multiple slave edges occur before a master edge, then only the net time of the last edge that was not ignored by "Slave leading" is stored. The "EdgeDetectSlavecount" on page 28 register can be used to detect multiple edges.

Data type	Value	Information
INT	-32,768 to 32,767	Net time of the slave edge (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Net time of the slave edge (32-bit)

11.11 Motion functions

Encoder emulation can be used to generate up/down counters (direction/frequency) and ABR encoder signals. The following conditions must be met to achieve an exact match of the position of the module with the remote station:

- Up/Down counter: The remote station must evaluate both rising and falling edges.
- ABR encoder: The remote station must employ 4x evaluation.

The motion function can be operated in 2 different modes:

- "Position control mode" on page 30
- "Speed control mode" on page 31

11.11.1 Position control mode

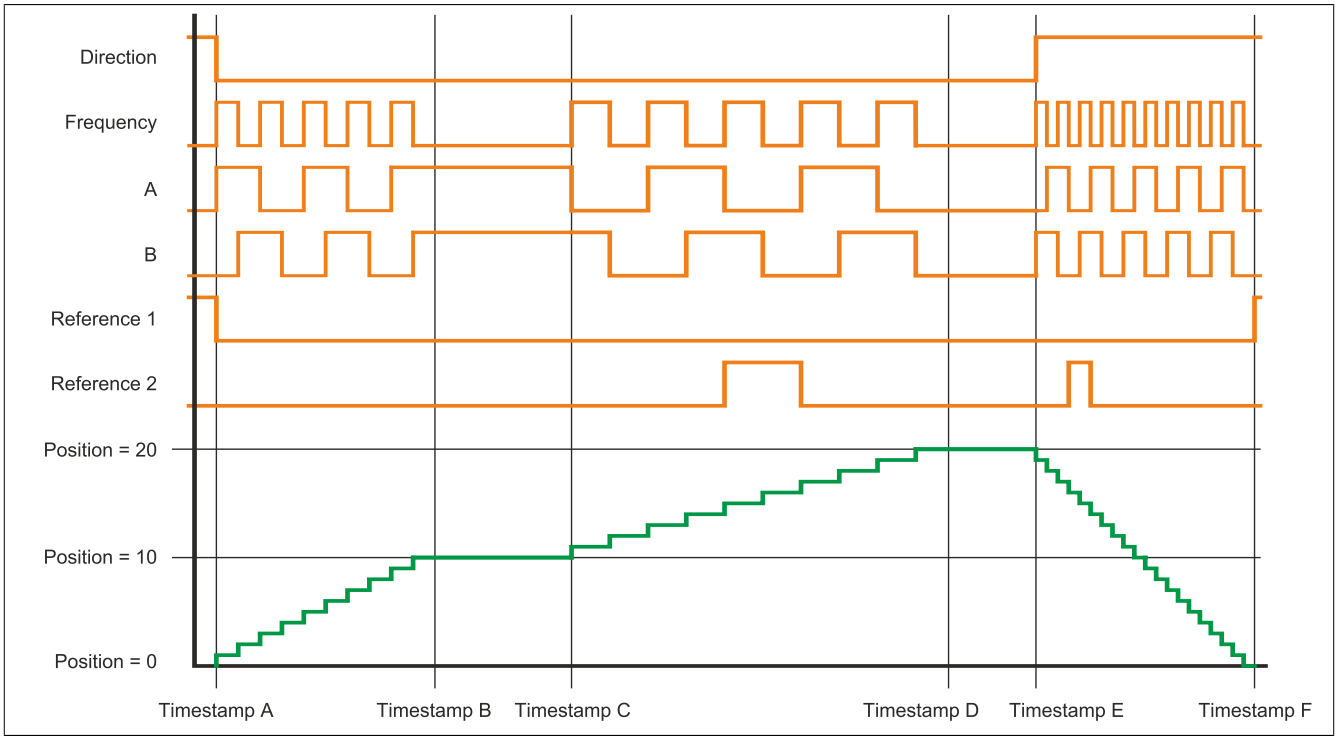
Each time the "MovTargetTime" on page 36 register changes, a new position setpoint is transferred from the "MovPosition" on page 36 register to the FIFO. The time/position data in the FIFO is then processed in such a manner that the positions are always reached at the time of the respective timestamps. This means that the module internally ensures that the positions are reached by the defined timestamps (number/frequency of the pulses is calculated automatically). The timestamps can be based on the X2X net time, the CPU's system time or the "MovCurrentTime" on page 37 register. Timestamps that are set in a manner that does not allow the required position change to be reached before the timestamp (output frequency of the pulse would exceed "CfO_SpeedLimit" on page 32) cause a MovMaxFrequencyViolation error.

Selected values for the example "Timing of movement":

Timestamp A = MovTimeValid + 40,000	Position for timestamp A = 0
Timestamp B = Timestamp A + 40.000	Position for timestamp B = 10
Timestamp C = Timestamp B + 25.000	Position for timestamp C = 10
Timestamp D = Timestamp C + 70,000	Position for timestamp D = 20
Timestamp E = Timestamp D + 15,000	Position for timestamp E = 20
Timestamp F = Timestamp E + 40,000	Position for timestamp F = 0

Configuration: Reference pulse 1 = Start position and margin, Start position = 0, Margin = 1

Configuration: Reference pulse 2 = Start and end position, Start position = 15, End position = 17



11.11.2 Speed control mode

In speed control mode, the application only specifies the speed setpoint. The module returns the current position in the "MovPosition (32-bit)" on page 37 register.

The internal timing is designed so that the value 16,777,216 (0x01000000) in the "MovSpeed" on page 37 register results in exactly one increment per "control period".

This creates the following relationship for 32-bit speed setpoints ("Data format of speed values = 32-bit):

$$MovSpeed = vOut * 2^{resol} * period$$

Unlike other registers, the 2 higher-value bytes of "MovSpeed (32-bit)" are set when the "MovSpeed (32-bit)" register is written. This creates the following relationship for the direct calculation with "MovSpeed (16-bit)"

$$MovSpeed = \frac{vOut * 2^{resol} * period}{2^{16}}$$

Variable	Description	Unit
MovSpeed	Value for "MovSpeed" register (16 or 32-bit)	
vOut	Desired output speed Each edge represents one increment (rising or falling).	Inc/s
resol	Value configured for the "CfO_ResolSpeed" on page 35 register	Bits
period	Value configured for the "CfO_SpeedCycleTime_32Bit" on page 34 register **Information:** **Must be set in µs in Automation Studio. The calculation is performed in s, however.**	s

11.11.3 Performing a movement in mode "Position control"

Several things must be kept in mind when operating the module in order to perform a movement without errors and avoid error messages.

Information:

The specified time/position pairs are not "movement commands", but position data that is continuously processed by the module.

- To allow the module to calculate movement pulses, the first time/position data pair (t, x) is interpreted as the home position. In this case, "t" represents the starting time point and "x" the current position. A movement is not yet performed.
- As long as "MovPosEnable" on page 36 is set to "True", time/position data pairs must be sent to the module continuously. As soon as the last data pair has been processed and the module has no more data pairs in the FIFO buffer, error message "MovFifoEmpty" is reported (see "Error state register - Movement functions" on page 13). Error message "MovTargetTimeViolation" is also reported since a "future point in time" for another movement was no longer found.
- To enable a standstill, the time/position data pairs must be specified with an unchanged position but future points in time.
- Setting "MovPosEnable" on page 36 to "False" immediately stops the current movement regardless of the current position or time specification.

11.11.4 "FifoSize" register

Name:
FifoSize
"Number of Fifo entries" in the Automation Studio I/O configuration.

Determines the size of the FIFO for "MovTargetTime" on page 36 and "MovTargetPosition" on page 36. One timestamp and one position that should be reached by the timestamp can be transferred to the FIFO per X2X cycle.

Data type	Value	Information
USINT	0	FIFO disabled
	3	8 entries (2^3)
	4	16 entries (2^4)
	5	32 entries (2^5)
	6	64 entries (2^6)
	7	128 entries (2^7)
	8	256 entries (2^8)

11.11.5 "CfO_Mode" register

Name:
CfO_Mode

This register can be used to configure the mode of the movement functions.

Data type	Values
USINT	See bit structure.

Bit structure:

Bit	Description	Value	Information
0	Must be enabled when working without timestamps. Enabled in Automation Studio if: • Movement = "speed control" • Movement = "position control and "data format / mode of preset time = local time"	0	Disabled
		1	Enabled
1	If this function is enabled, then a new positioning movement is triggered as soon as the value changes in the "MovPosition" on page 36 register. Enabled in Automation Studio if: • Movement = "position control and "data format / mode of preset time = local time"	0	No position control (speed control)
		1	Position control enabled (position control)
2	Reference mode 1 "Configuration reference pulse 1" in the Automation Studio I/O configuration.	0	Start/end position
		1	Start position and span
3	Reference mode 2 "Configuration reference pulse 1" in the Automation Studio I/O configuration.	0	Start/end position
		1	Start position and span
4 - 7	Reserved	-	

11.11.6 "CfO_SpeedLimit" register

Name:
CfO_SpeedLimit
"Max. movement frequency" in the Automation Studio I/O configuration.

Configures the maximum permitted output frequency and the maximum internal computing frequency. The higher internal computing frequencies of 500 kHz and 2, 4, 8, 16, 32 and 64 MHz can only be achieved by configuring n bits as decimal places (see "CfO_ResolPosition" on page 34 register).

Data type	Value	Max. increment frequency	Max. frequency for frequency output channel	Max. frequency for A/B output channel
USINT	253	64 MHz	125 kHz	625 kHz
	254	32 MHz		
	255	16 MHz		
	0	8 MHz		
	1	4 MHz		
	2	2 MHz		
	3	1 MHz		
	4	500 kHz		
	5	250 kHz (default)		
	6	125 kHz	625 kHz	3125 kHz

Information:

In "position control" mode, the increment frequencies 16, 32 and 64 MHz are not permitted to be used when a 29-bit timestamp is set (see "CfO_TimeStampRange" on page 33 register) due to an internal range violation.

11.11.7 "CfO_FormatAdjust" register

Name:
CfO_FormatAdjust

This register determines the number of absolute bits that can be output on the signal output (With a direction/frequency signal, the bit with the lowest value can be output directly on the frequency output. With an AB signal, 2 bits are possible.)

Data type	Value	Information
USINT	1 to 2	Number of absolute bits (Automation Studio default = 1)

11.11.8 "CfO_TimeStampRange" register

Name:
CfO_TimeStampRange
"Data format/mode of target time value" in the Automation Studio I/O configuration.

The width of the transferred timestamp data in the module is configured in this register.

Information:

Because the module uses an internal resolution of 1/8 µs, timestamp data is processed internally at a maximum width of 29 bits.

Data type	Value	Information
SINT	16	16-bit timestamp ("16-bit" selected in the Automation Studio I/O configuration)
	24	24-bit timestamp ("local time" or "speed control" movement selected in the Automation Studio I/O configuration)
	29	29-bit timestamp ("29-bit" selected in the Automation Studio I/O configuration)

11.11.9 "CfO_PositionsRange" register

Name:
CfO_PositionsRange
"Target position range" in the Automation Studio I/O configuration.

The number of bits for position control are configured in this register. The "PositionRange" must be reduced if, for example, the movement function should follow the absolute value of a 12-bit SSI encoder. In this case, the bit width of the movement position also has to be limited to the number of bits of the encoder, or else the movement position would not also overrun if the encoder were to overrun. In this case, the module would attempt (in the opposite direction) to reach the position of an encoder that had just overrun.

Example

The 12-bit SSI encoder overruns from 2047 to -2048. The module would generate 4096 negative increments if more than 12 bits were defined for "CfO_PositionRange" in order to reach position -2048 from the position 2047.

Information:

If the 16-bit value of the "MovPosition" on page 37 register is used, then the bit width of the position must also be limited to ≤16 bits or else this would also result in incorrect overrun behavior.

Data type	Value	Information
SINT	8 to 32	Number of bits for position control

11.11.10 "CfO_ReferenceRange" register

Name:
CfO_Reference0Range to CfO_Reference1Range
"Reference#1 range" to "Reference#2 range" in the Automation Studio I/O configuration.

This register determines the number of bits that can be used for the reference position comparison. This makes it possible to generate a reference pulse every 2^n increments.

Information:

The number of bits set in this register must not be higher than the number of bits set for "MovReferenceStart" on page 35 and "MovReferenceStopMargin" on page 35.

Data type	Value	Information
SINT	4 to 32	Number of bits for position comparison

11.11.11 "CfO_TimeStampDelay" register

Name:
CfO_TimeStampDelay
"Target time delay" in the Automation Studio I/O configuration.

All timestamps are delayed by the value defined in this register.

Information:

When setting to "Local time" in register "CfO_TimeStampRange" on page 33, a value at least 2x the X2X cycle time in µs must be entered.

Data type	Value	Information
DINT	0 to 1000000	Timestamp delay in µs

11.11.12 "CfO_SpeedCycleTime_32Bit" register

Name:
CfO_SpeedCycleTime_32Bit
"Control period" in the Automation Studio I/O configuration.

The control period for "speed control" mode can be set in this register in steps of 1/8 µs.

Information:

The value defined in the Automation Studio I/O configuration under "Control period" is automatically multiplied by 8 and then used as CfO_SpeedCycleTime_32bit.

Data type	Value	Information
UDINT	400 to 40000	Control period for "speed control" mode

11.11.13 "CfO_ResolPosition" register

Name:
CfO_ResolPosition
"Position resolution" in the Automation Studio I/O configuration.

This register contains the number of bits as decimal place for jitter reduction. Internally, the module adds 2^n (n = number of decimal places) to the frequency, which results in edge switching times with a higher resolution. The output switching frequency is not increased from a hardware perspective, but the edge timing is more precise.

Data type	Value	Information
SINT	0	Default, no decimal places
	1 to 14	Selection of bits as decimal places

Information:

Keep in mind that each configured decimal place also limits the maximum number range by that number of bits.

For example: 0 decimal places →maximum position range = 29 bits

3 decimal places →maximum position range = 26 bits

Also keep in mind that the "CfO_SpeedLimit" on page 32 register must be adjusted for these higher computing frequencies based on the number of configured decimal places.

11.11.14 "CfO_ResolSpeed" register

Name:
CfO_ResolSpeed
"Speed resolution" in the Automation Studio I/O configuration.

This register contains the number of bits as decimal place for jitter reduction of the speed value. Internally, the module adds 2^n (n = number of decimal places) to the frequency, which results in edge speed values with a higher resolution.

Due to the bit limitation, a 16 or 32-bit speed value is set in the Automation Studio I/O configuration. Since the internal calculation is always based on 32-bit, when configured to 16-bit an offset of 16 must always be added to the desired number of decimal places.

Data type	Value	Information
SINT	0 to 31	Selection of bits as decimal places; Bus controller default: 24

Information:

Keep in mind that each configured decimal place also limits the maximum number range by that number of bits.

11.11.15 "CfO_ReferenceStart / MovReferenceStart" register

Name:
CfO_Reference0Start to CfO_Reference1Start
MovReference1Start to MovReference2Start
"Start position" in the Automation Studio I/O configuration

The start position for the reference pulse is shown in these registers.

In the positive direction, the output (R) is set when the start position is reached. In negative direction, the output is reset as soon as the value falls below the start position value.

Data type	Value	Information
INT	-32,768 to 32,767	Start position (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Start position (32-bit)

11.11.16 "CfO_ReferenceStopMargin / MovReferenceStopMargin" register

Name:
CfO_Reference0StopMargin to CfO_Reference1StopMargin
MovReference1StopMargin to MovReference2StopMargin
"End position or margin" in the Automation Studio I/O configuration

The end position or the margin in which the reference pulse is output is configured in these registers.

If "Reference mode x = Start/end position" is used in the "CfO_Mode" on page 32 register, then the output (R) is reset when the end position is reached in the positive direction. In the negative direction, the output is set as soon as the value falls below the end position value.

When "Reference mode x = Start position and span", the content of this register is added to the start position and the resulting sum is used as the end position.

Data type	Value	Information
INT	-32,768 to 32,767	End position (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	End position (32-bit)

11.11.17 "CfO_AccelDataInit / MovAcceleration" register

Name:
CfO_AccelDataInit
MovAcceleration
"Acceleration value" in the Automation Studio I/O configuration.

This register shows the acceleration value in increments per control period².

- 32-bit: 16777216 (0x01000000) corresponds to 1 increment per control period²
- 16-bit: 256 (0x0100) corresponds to 1 increment per control period²

Data type	Value	Information
UINT	0 to 65,535	Acceleration value (16-bit)
UDINT	0 to 4,294,967,296	Acceleration value (32-bit)

11.11.18 "MovementControl" register

Name:
MovPosEnable
MovSpeedEnable

This register can be used to enable position and speed control.

Data type	Values
USINT	See bit structure.

Bit structure:

Bit	Description	Value	Information
0	MovPosEnable	0	Position control disabled
		1	Position control enabled
1	MovSpeedEnable	0	Speed control disabled
		1	Speed control enabled
2 - 6	Reserved	-	
7	Movement reset (immediate stop)	0	Passive reset
		1	Active reset

11.11.19 "MovTargetTime" register

Name:
MovTargetTime

Timestamp data is shown in this register. Each time this register changes, the new position data ("MovTargetPosition" on page 36) and timestamp data are transferred to the FIFO. When "MovSpeedEnable = True", the module calculates the output speed (frequency) so that the "MovTargetPosition" is reached at "MovTargetTime".

Data type	Value	Information
INT	-32,768 to 32,767	Timestamp (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Timestamp (32-bit)

Information:

Only 29 bits of this register are processed internally.

11.11.20 "MovTargetPosition" register

Name:
MovTargetPosition

Position data is shown in this register.

Data type	Value	Information
INT	-32,768 to 32,767	Position (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Position (32-bit)

11.11.21 "MovSpeed" register

Name:
MovSpeed

This register shows the speed setpoint for "speed control" mode in increments per control period.

- 32-bit: 16,777,216 (0x01000000) corresponds to 1 increment per control period
- 16-bit: 256 (0x0100) corresponds to 1 increment per control period

Data type	Value	Information
INT	-32,768 to 32,767	Speed setpoint (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Speed setpoint (32-bit)

11.11.22 "MovTimeValid" register

Name:
MovTimeValid

This register displays the net time of the current position.

Data type	Value	Information
INT	-32,768 to 32,767	Net time of the current position (16-bit).
DINT	-2,147,483,648 to 2,147,483,647	Net time of the current position (32-bit)

11.11.23 "MovPosition" register

Name:
MovPosition

This register shows the current position.

Data type	Value	Information
INT	-32,768 to 32,767	Current position (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Current position (32-bit)

11.12 Synchronous serial interface (SSI)

The synchronous serial interface makes it possible to receive data from SSI absolute encoders.

Two lines are needed for data exchange:

SSI clock:	Generated by the module on output 7 (if configured).
SSI data:	A data bit is transferred from the encoder to the module with each clock pulse (input 5 can be used as the SSI input).

11.12.1 SSI transfer process

When the first edge occurs on the SSI clock, a monoflop is triggered in the encoder and the current parallel pending value is latched to the offset register (the low level of the monoflop prevents other values from being added to the offset register during data transfer).

The highest value bit is then transferred to the module when the next edge occurs.

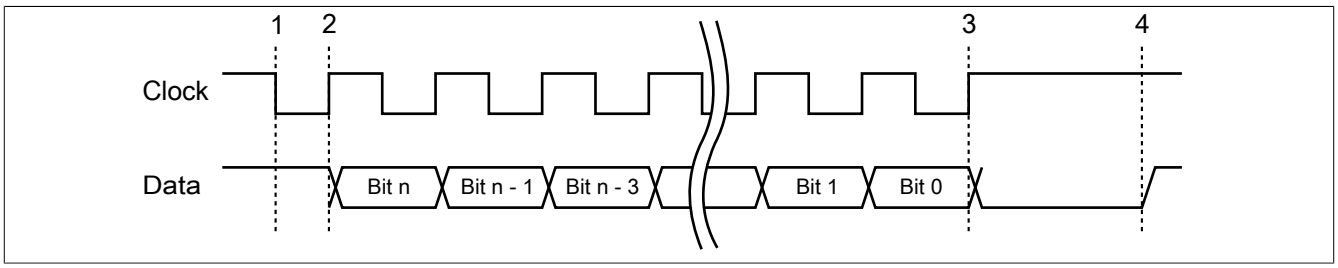
With each subsequent cycle, the next lowest bit is transferred. The cycles re-trigger the monoflop constantly so that its output prevents new data from being accepted.

The sequence of cycles stops once the number of data bits defined in the "CfO_DataBits" on page 39 register has been received.

The monoflop is no longer triggered. After a certain amount of time has passed (depending on the encoder), the monoflop's output re-assumes the output level, thereby enabling parallel data to be accepted once again in the encoder's offset register.

When the "Monoflop check" is run, the data line is queried for the configured level before a new transfer is started. This makes it possible to ensure that the monoflop really has reset before a new transfer is started.

Transfer to synchronous serial interface



Measurement value processing

1 Starting bit ... Stores the measurement value
2 Output of first data bit
3 All data bits are transferred, monostable multivibrator time starts counting down.
4 Monostable multivibrator returns to its initial state. A new transfer can be started.

11.12.2 "CfO_CycleSelect" register

Name:
CfO_CycleSelect
"Update cycle" in the Automation Studio I/O configuration.

SSI transfer is started at the update cycle. The clock sequence is generated on the SSI clock output. The first edge of the clock signal triggers the monoflop in the encoder and latches the current position. At the same time, the current net time is also recorded in the "SSITimeValid" on page 39 register. As soon as all bits have been transferred via the SSI, the position is passed on with the next "SIframeGenCycle" via the X2X Link. A SSICycleTimeViolation error is reported if the SSI transfer is not completed within the SSI update cycle (e.g. system timer as update cycle). The SSI transfer is still fully completed and then started again with the next update cycle.

Data type	Value	Information
USINT	2	System timer
	3	Prescaled system timer
	10	AOAI
	14	SOSI

11.12.3 "CfO_PhysicalMode" register

Name:
CfO_PhysicalMode
"Parity bit" in the Automation Studio I/O configuration
"Monoflop check" in the Automation Studio I/O configuration
"Data format" in the Automation Studio I/O configuration
"Clock frequency" in the Automation Studio I/O configuration

The SSI interface is configured in this register.

Data type	Values
USINT	See bit structure.

Bit structure:

Bit	Description	Value	Information
0 - 1	"Parity bit"[1]	00	Disabled
		01	Even parity
		10	Uneven parity
		11	Ignored (the parity bit is transferred, but not evaluated).
2 - 3	"Monoflop check"[2]	00	Disabled
		01	Low level (data signal is checked for low level after the monoflop has reset).
		10	High level (data signal is checked for high level after the monoflop has reset).
		11	Ignored (the necessary cycle is triggered, but not evaluated).
4	"Data format"	0	Encoder with binary output
		1	Encoder with Gray Code. The module converts the position data into binary format.
5	Reserved	-	
6 - 7	"Clock frequency"	00 to 10	Not permitted
		11	125 kHz

1) If the parity bit does not match, then a SSIParityError is generated and the position data is not accepted in the "SSIPosition" on page 40 register.
2) A new SSI transfer is not started until the data signal has assumed the level defined for the "monoflop check" after the transfer. This then triggers the error SSICycleTimeViolation.

11.12.4 "CfO_DataBits" register

Name:
CfO_DataBits
"Valid SSI bit length" in the Automation Studio I/O configuration.

Determines the number of valid data bits to be transferred via the SSI. The valid data bits are used for the "SSI-Position" on page 40.

Data type	Value	Information
USINT	1 to 32	Number of valid data bits

11.12.5 "CfO_NullBits" register

Name:
CfO_NullBits
"Leading zero bits" in the Automation Studio I/O configuration.

This register can be used to configure the number of leading zero bits. The leading zero bits can be required before the valid data bits.

Data type	Value	Information
USINT	0 to 31	Number of leading zero bits

11.12.6 "SSITimeValid" register

Name:
SSITimeValid

This register displays the net time of the current position.

Data type	Value	Information
INT	-32,768 to 32,767	Net time of the current position (16-bit).
DINT	-2,147,483,648 to 2,147,483,647	Net time of the current position (32-bit)

11.12.7 "SSITimeChanged" register

Name:
SSITimeChanged

The net time of the last position change is shown in this register.

Data type	Value	Information
INT	-32,768 to 32,767	Net time of the last position change (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Net time of the last position change (32-bit)

11.12.8 "SSIPosition" register

Name:
SSIPosition

This register shows the current position sent via the SSI interface.

Data type	Value	Information
INT	-32,768 to 32,767	Current position (16-bit)
UDINT	0 to 4,294,967,295	Current position (32-bit)
DINT	-2,147,483,648 to 2,147,483,647	

11.13 Counter

The universal counter pair can be used in 3 different modes. Here, signals up to 100kHz are reliably measured. Up to 4 latch inputs can be configured in all modes. Enabled latch inputs are negated if necessary and connected with a logical AND operation for a latch condition. If the latch condition is met, the current counter value is stored in a separate register.

Inputs

The physical inputs have fixed assignments based on the respective mode.

Mode	Input 1	Input 2	Input 5	Input 6
Edge counters	Counter input for counter 1 Latch input 1	Counter input for counter 2 Latch input 2	- Latch input 3	- Latch input 4
Up/down counter	Counting direction Latch input 1	Counter frequency Latch input 2	- Latch input 3	- Latch input 4
Incremental encoder	A Latch input 1	B Latch input 2	- Latch input 3	- Latch input 4

Latch function

As latch inputs, inputs 1, 2, 5, and 6 can each be polled to determine if they have a HIGH or LOW level.

In "Latch mode = continuous", the counters are latched once as soon as "LatchEnable = TRUE" and the configured latch condition is met. If the latch condition is met again, then the counter value is also latched again. (i.e. One latch event is triggered with each rising edge on the output of the AND operation of all latch inputs).

In "Latch mode = single", the counters are latched once as soon as "LatchEnable = TRUE" and the configured latch condition is met. If the latch condition is met again, then the counter value is not automatically copied again. Another latch event can only be processed after "LatchEnable = False" and then "LatchEnable = True" again.

11.13.1 "CfO_CounterCycleSelect" register

Name:
CfO_CounterCycleSelect
"Update cycle" in the Automation Studio I/O configuration.

The update cycle for the counter values is configured in this register.

> **Information:**
>
> **The maximum counting frequency depends on the cycle. The module can process a maximum of 200 increments (edges) within a counter cycle.**

Data type	Value	Information
USINT	2	System timer
	3	Prescaled system timer
	10	AOAI time of X2X cycle
	14	SOSI time of X2X cycle

11.13.2 "CfO_CounterMode" register

Name:
CfO_CounterMode
"Counter mode" in the Automation Studio I/O configuration.

The counter mode is configured in this register.

Data type	Value	Information
USINT	0	Edge counters In this mode, the two counters are used as edge counters. The counter input of counter 1 is linked permanently to input 1 and the counter input of the second counter is linked permanently to input 2. Both rising as well as falling edges are counted.
	2	Up/down counter The up/down counter works according to the direction/frequency principle. Input 1 determines the counting direction (LOW = positive, HIGH = negative), input 2 serves as the counting frequency input. Both rising as well as falling edges on the counting frequency input are counted.
	3	Incremental encoder (AB counter) When configured as an AB counter, input 1 serves as the A channel and input 2 as the B channel. All edges are evaluated (4x evaluation).

11.13.3 "CfO_LatchMode" register

Name:
CfO_LatchMode
"Latch mode" in the Automation Studio I/O configuration.

The latch mode is configured in this register.

Data type	Values
USINT	See bit structure.

Bit structure:

Bit	Description	Value	Information
0	LatchMode	0	Single shot
		1	Continuous
1 - 7	Reserved	-	

11.13.4 "CfO_LatchComparator" register

Name:
CfO_LatchComparator

"Latch level channel 0x" in the Automation Studio I/O configuration.

The latch comparators for the counter inputs are configured in this register.

Data type	Values
USINT	See bit structure.

Bit structure:

Bit	Description	Value	Information
0	Comparison level for latch comparator on input 1	0	LOW
		1	HIGH
1	Comparison level for latch comparator on input 2	0	LOW
		1	HIGH
2	Comparison level for latch comparator on input 5	0	LOW
		1	HIGH
3	Comparison level for latch comparator on input 6	0	LOW
		1	HIGH
4	Enable latch comparator on input 1	0	Disabled
		1	Enabled
5	Enable latch comparator on input 2	0	Disabled
		1	Enabled
6	Enable latch comparator on input 5	0	Disabled
		1	Enabled
7	Enable latch comparator on input 6	0	Disabled
		1	Enabled

11.13.5 "CounterControl" register

Name:
CounterReset
LatchEnable

This register can be used to clear counter values or enable the latch.

Data type	Values
USINT	See bit structure.

Bit structure:

Bit	Description	Value	Information
0	CounterReset	0	No action
		1	Delete counter
1	LatchEnable	0	Disabled
		1	Enabled
2 - 7	Reserved	-	

11.13.6 "LatchCount" register

Name:
LatchCount

Latch events are counted in this register. This counter can be used to detect whether a new value has been latched.

Data type	Value	Information
SINT	-128 to 127	Latch counter

11.13.7 "CounterTimeValid" register

Name:
CounterTimeValid

This register displays the X2X net time of the current counter value.

Data type	Value	Information
INT	-32,768 to 32,767	Net time of the current counter value (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Net time of the current counter value (32-bit)

11.13.8 "CounterTimeChanged" register

Name:
Counter01TimeChanged to Counter02TimeChanged

The net time of the last change to the respective counter is shown in this register.

Data type	Value	Information
INT	-32,768 to 32,767	Net time of the last change to the respective counter (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Net time of the last change to the respective counter (32-bit)

11.13.9 "CounterValue" register

Name:
CounterValue01 to CounterValue02

This register shows the current value of the respective counter.

Data type	Value	Information
INT	-32,768 to 32,767	Value of the respective counter (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Value of the respective counter (32-bit)

11.13.10 "CounterLatch" register

Name:
CounterLatch01 to CounterLatch02

As soon as the latch conditions defined in the "CfO_LatchComparator" on page 41 register have been met, the contents of the respective "CounterValue" on page 42 register are copied to this register.

Data type	Value	Information
INT	-32,768 to 32,767	Latch counter (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Latch counter (32-bit)

11.13.11 "CounterRel" register

Name:
CounterRel01 to CounterRel02

The value of the respective counter, relative to the last latch of the respective counter is calculated in this register.

Data type	Value	Information
INT	-32,768 to 32,767	Counter value relative to the last latch (16-bit)
DINT	-2,147,483,648 to 2,147,483,647	Counter value relative to the last latch (32-bit)

11.14 Minimum X2X cycle time

The minimum X2X cycle time is strongly dependent on the configured functions and the resulting load on the module. A "Fast reaction" setting and very short system cycle (<50 µs) generally have a negative effect on the minimum X2X cycle time. This can lead to errors when the X2X cycle time is short.