# X67DC1198

## 1 General information

The possibilities are almost endless with this digital counter module.

Connectors 1 and 3 are 12-pin M12 connectors. Each of these can be used to connect 1 incremental encoder or SSI encoder with 5 V differential signals. In addition, there are 2 digital channels available on the same output which, when configured as inputs, can be used with incremental encoders with status outputs (e.g. alarms). When configured as outputs, they act as preset and count direction switching functions when used together with SSI encoders, for example.

2 more sockets and 2 more configurable digital channels are available on female connectors 2 and 4. The inputs can be used as latch, gate, or home enable switches, while the outputs can be used as comparator outputs, for example.

- 2 incremental or SSI encoder inputs 5 V
- 2 digital channels, 24 V per connection, configurable either as inputs or outputs
- 4 AB counters on the digital inputs
- Pulse width modulation of the digital outputs
- Encoder power supply 5 V and 24 V integrated in encoder connection

### Information:

**In contrast to freely configurable function model "Standard", the function selection cannot be modified in function model "Bus controller".**

**Function model "Bus Contoller":**

- **1x ABR incremental encoder (5 V)**
- **1x SSI absolute encoder (5 V)**
- **1x PWM output (24 V)**
- **1x up/down counter (24 V)**
- **3x AB counter (24 V)**

## 2 Order data

| Model number | Short description | Figure |
|---|---|---|
| | **Multi-function** |   |
| X67DC1198 | X67 digital counter module, 2x 3 inputs 5 V for SSI 1 Mbit/s or ABR 250 kHz, 8 digital channels 24 VDC, 0.1 A, configurable as inputs or outputs or 4 AB counters 100 kHz or 4x comparator outputs or 2x PWM outputs, local time measurement functions | |

Table 1: X67DC1198 - Order data

| Required accessories |
|---|
| For a general overview, see section "Accessories - General overview" of the X67 system user's manual. |

# 3 Technical data

| Model number | X67DC1198 |
|---|---|
| **Short description** | |
| I/O module | 2 SSI absolute encoders 5 V or 2 ABR incremental encoders 5 V,<br>4 AB counters or 4 up/down counters 24 V,<br>2x pulse width modulation, time measurement, relative timestamp |
| **General information** | |
| Isolation voltage between encoder and bus | 500 $V_{Eff}$ |
| B&R ID code | 0x18D0 |
| Sensor/Actuator power supply | 0.5 A summation current |
| Status indicators | I/O function for each channel, supply voltage, bus function |
| Diagnostics | |
| Outputs | Yes, using status LED and software |
| I/O power supply | Yes, using status LED and software |
| Connection type | |
| X2X Link | M12, B-keyed |
| Inputs/Outputs | 2x M12, 5-pin, A-keyed |
| SSI/ABR encoder | 2x M12, 12-pin, A-keyed |
| I/O power supply | M8, 4-pin |
| Power consumption | |
| Internal I/O | 2.8 W |
| X2X Link power supply | 0.75 W |
| Certifications | |
| CE | Yes |
| KC | Yes |
| EAC | Yes |
| UL | cULus E115267<br>Industrial control equipment |
| HazLoc | cCSAus 244665<br>Process control equipment<br>for hazardous locations<br>Class I, Division 2, Groups ABCD, T5 |
| ATEX | Zone 2, II 3G Ex nA IIA T5 Gc<br>IP67, Ta = 0 - Max. 60°C<br>TÜV 05 ATEX 7201X |
| **I/O power supply** | |
| Nominal voltage | 24 VDC |
| Voltage range | 18 to 30 VDC |
| Integrated protection | Reverse polarity protection |
| Power consumption | |
| Sensor/Actuator power supply | Max. 12 W [1)] |
| **Sensor/Actuator power supply** | |
| Voltage | I/O power supply minus voltage drop for short circuit protection |
| Voltage drop for short-circuit protection at 0.5 A | Max. 2 VDC |
| Summation current | Max. 0.5 A |
| Short-circuit proof | Yes |
| **SSI absolute encoder** | |
| Quantity | 2 |
| Encoder inputs | 5 V, symmetrical |
| Counter size | 32-bit |
| Max. transfer rate | 1 Mbit/s |
| Coding | Gray/Binary |
| Overload characteristics of encoder power supply | Short circuit protection, overload protection |
| Transfer rate | 125 kbit/s, 250 kbit/s, 500 kbit/s, 1 Mbit/s |
| Encoder power supply | |
| 5 VDC | Module-internal, max. 0.3 A summation current |
| 24 VDC | Module-internal, max. 0.5 A summation current |
| **ABR incremental encoder** | |
| Quantity | 2 |
| Encoder inputs | 5 V, symmetrical |
| Counter size | 16/32-bit |
| Input frequency | Max. 250 kHz |
| Evaluation | 4x |
| Encoder power supply | |
| 5 VDC | Module-internal, max. 0.3 A summation current |
| 24 VDC | Module-internal, max. 0.5 A summation current |
| Input filter | |
| Hardware | ≤200 ns |
| Software | - |
| Common-mode range | -7 V ≤ $V_{CM}$ ≤ +12 V |
| Overload characteristics of encoder power supply | Short circuit protection, overload protection |
| **AB counters** | |
| Quantity | 4 |

Table 2: X67DC1198 - Technical data

| Model number | X67DC1198 |
|---|---|
| Evaluation | 4x |
| Input frequency | Max. 100 kHz |
| Encoder inputs | 24 V, asymmetrical |
| Encoder power supply 24 VDC | Module-internal, max. 0.5 A summation current |
| Counter size | 16/32-bit |
| **Digital inputs 5 VDC** | |
| Quantity | Up to 6, configurable as inputs or outputs using software |
| Nominal voltage | 5 VDC differential signal, EiA RS-485 standard |
| Input characteristics per EN 61131-2 | Type 1 |
| Input filter | |
| Hardware | 200 ns |
| Software | - |
| Additional functions | ABR incremental encoder, SSI absolute encoder, event counting, time measurement, relative timestamp |
| **Digital inputs 24 VDC** | |
| Quantity | Up to 8, configurable as inputs or outputs using software |
| Nominal voltage | 24 VDC |
| Input characteristics per EN 61131-2 | Type 1 |
| Input circuit | Sink |
| Input voltage | 18 to 30 VDC |
| Input current at 24 VDC | Approx. 3.3 mA |
| Input resistance | 7.31 kΩ |
| Input filter | |
| Hardware | ≤2 µs |
| Software | - |
| Switching threshold | |
| Low | <5 VDC |
| High | >15 VDC |
| Additional functions | Reference enable inputs for ABR, event counting, latch function, time measurement, relative timestamp |
| **Event counter** | |
| Quantity | 8 |
| Evaluation | 2x |
| Input frequency | Max. 100 kHz |
| Encoder inputs | 24 V, asymmetrical |
| Encoder power supply 24 VDC | Module-internal, max. 0.5 A summation current |
| Counter size | 16/32-bit |
| **Up/Down counters** | |
| Quantity | 4 |
| Evaluation | 2x |
| Input frequency | Max. 100 kHz |
| Encoder inputs | 24 V, asymmetrical |
| Encoder power supply 24 VDC | Module-internal, max. 0.5 A summation current |
| Counter size | 16/32-bit |
| **Edge detection / Time measurement** | |
| Possible measurements | Gate time, period duration, edge offset for various channels |
| Measurements per module | Up to 9 |
| Measurements per channel | Up to 2 |
| Counter size | 16-bit |
| Counter frequency | |
| Internal | 8 MHz, 4 MHz, 2 MHz, 1 MHz, 500 kHz, 250 kHz, 125 kHz, 62.5 kHz |
| Signal form | Square wave pulse |
| Measurement type | Continuous or triggered |
| **Digital outputs 5 VDC** | |
| Quantity | Up to 6, configurable as inputs or outputs using software |
| Type | 5 VDC differential signal, EiA RS-485 standard |
| Output circuit | Sink or source |
| Output protection | Short circuit protection |
| Variant | Push / Pull / Push-Pull |
| Diagnostic status | Output monitoring |
| **Digital outputs 24 VDC** | |
| Quantity | Up to 8, configurable as inputs or outputs using software |
| Nominal voltage | 24 VDC |
| Nominal output current | 0.1 A |
| Total nominal current | 0.8 A |
| Variant | Push / Pull / Push-Pull |
| Output circuit | Sink or source |
| Output protection | Thermal cutoff for overcurrent and short circuit, integrated protection for switching inductances, reverse polarity protection for output power supply |
| Braking voltage when switching off inductive loads | Switching voltage + 0.6 VDC |
| Diagnostic status | Output monitoring |
| Switch-on in the event of overload shutdown or short-circuit shutdown | Approx. 10 ms (depends on the module temperature) |
| Peak short-circuit current | <10 A |
| Leakage current when switched off | Max. 25 µA |

Table 2: X67DC1198 - Technical data

| Model number | X67DC1198 |
|---|---|
| Residual voltage | <0.9 V at 0.1 A nominal current |
| Switching voltage | I/O power supply minus residual voltage |
| Pulse width modulation [2] | |
| Period duration | 41.6 µs to 500 ms |
| Pulse duration | 0 to 100% |
| Resolution | 0.1% |
| Switching frequency | |
| Resistive load | Max. 24 kHz |
| Switching delay | |
| 0 → 1 | <2 µs |
| 1 → 0 | <2 µs |
| Additional functions | Pulse width modulation, comparator function |
| **Electrical properties** | |
| Electrical isolation | Bus isolated from encoder and channel<br>Channel not isolated from channel and encoder<br>Encoder not isolated from encoder |
| **Operating conditions** | |
| Mounting orientation | |
| Any | Yes |
| Installation elevation above sea level | |
| 0 to 2000 m | No limitations |
| >2000 m | Reduction of ambient temperature by 0.5°C per 100 m |
| Degree of protection per EN 60529 | IP67 |
| **Ambient conditions** | |
| Temperature | |
| Operation | -25 to 60°C |
| Derating | - |
| Storage | -40 to 85°C |
| Transport | -40 to 85°C |
| **Mechanical properties** | |
| Dimensions | |
| Width | 53 mm |
| Height | 85 mm |
| Depth | 42 mm |
| Weight | 200 g |
| Torque for connections | |
| M8 | Max. 0.4 Nm |
| M12 | Max. 0.6 Nm |

Table 2: X67DC1198 - Technical data

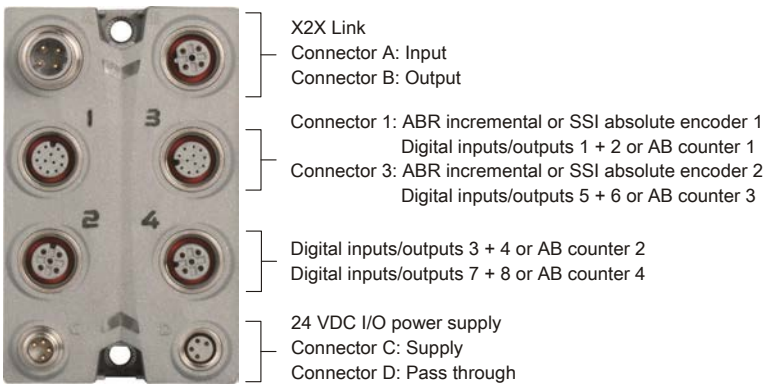
1) The power consumption of the sensors and actuators connected to the module is not permitted to exceed 12 W.
2) Dead time when switching between push and pull: max. 1.5 µs.

# 4 LED status indicators

<table>
<tr><th>Figure</th><th>LED</th><th colspan="3">Description</th></tr>
<tr><td rowspan="15"> 
</td><td>Status indicator 1</td><td colspan="3">Status indicator - X2X Link.</td></tr>
<tr><td></td><td>Green</td><td>Red</td><td>Description</td></tr>
<tr><td></td><td>Off</td><td>Off</td><td>No power supply via X2X Link</td></tr>
<tr><td></td><td>On</td><td>Off</td><td>X2X Link supplied, communication OK</td></tr>
<tr><td></td><td>Off</td><td>On</td><td>X2X Link supplied but X2X Link communication not functioning</td></tr>
<tr><td></td><td>On</td><td>On</td><td>PREOPERATIONAL: X2X Link supplied, module not initialized</td></tr>
<tr><td>1 - 4</td><td colspan="3">Status indicator for input/output 1 to 8</td></tr>
<tr><td></td><td colspan="2">LED</td><td>Description</td></tr>
<tr><td></td><td colspan="2">Orange</td><td>Output status of channel x</td></tr>
<tr><td></td><td colspan="2">Green</td><td>Input status of channel x</td></tr>
<tr><td></td><td colspan="2">Orange and green</td><td>One I/O channel is configured as an input and one as an output. Both channels are active. Both the orange and green LED are lit. Since only one light conductor is used, however, the color is mixed.</td></tr>
<tr><td>Status indicator 2</td><td colspan="3">Status indicator for module function.</td></tr>
<tr><td></td><td>LED</td><td>Status</td><td>Description</td></tr>
<tr><td></td><td rowspan="5">Green</td><td>Off</td><td>No power to module</td></tr>
<tr><td></td><td>Single flash</td><td>RESET mode</td></tr>
<tr><td></td><td>Blinking</td><td>PREOPERATIONAL mode</td></tr>
<tr><td></td><td>Double flash</td><td>BOOT mode (during firmware update)[1)]</td></tr>
<tr><td></td><td>On</td><td>RUN mode</td></tr>
<tr><td></td><td rowspan="4">Red</td><td>Off</td><td>No power to module or everything OK</td></tr>
<tr><td></td><td>On</td><td>Error or reset status</td></tr>
<tr><td></td><td>Single flash</td><td>Warning/Error on an I/O channel. Level monitoring for digital outputs has been triggered.</td></tr>
<tr><td></td><td>Double flash</td><td>Supply voltage not in the valid range</td></tr>
</table>

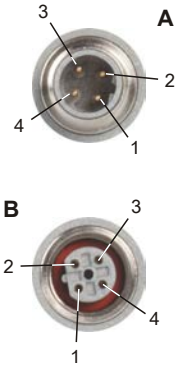
1) Depending on the configuration, a firmware update can take up to several minutes.

# 5 Connection elements




# 6 X2X Link

This module is connected to X2X Link using pre-assembled cables. The connection is made using M12 circular connectors.

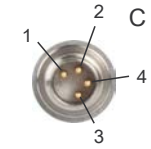
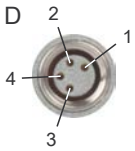
<table>
<tr><th>Connection</th><th colspan="2">Pinout</th></tr>
<tr><td rowspan="7"> 
</td><td>Pin</td><td>Description</td></tr>
<tr><td>1</td><td>X2X+</td></tr>
<tr><td>2</td><td>X2X</td></tr>
<tr><td>3</td><td>X2X⊥</td></tr>
<tr><td>4</td><td>X2X\</td></tr>
<tr><td colspan="2">Shield connection made via threaded insert in the module.</td></tr>
<tr><td colspan="2">A → B-keyed (male), input<br>B → B-keyed (female), output</td></tr>
</table>

## 7 24 VDC I/O power supply

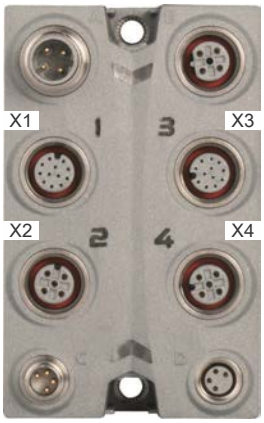
The I/O power supply is connected via M8 connectors C and D. The I/O power supply is connected via connector C (male). Connector D (female) is used to route the I/O power supply to other modules.

### Information:

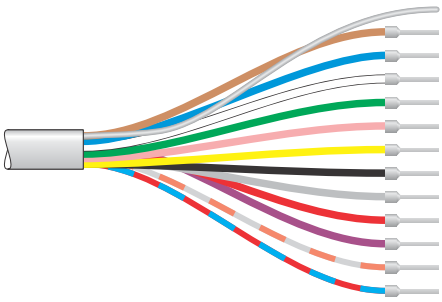
**The maximum permissible current for the I/O power supply is 8 A (4 A per connection pin)!**

| Connection | Pinout | |
|---|---|---|
|  | **Pin** | **Description** |
| | 1 | 24 VDC |
| | 2 | 24 VDC |
| | 3 | GND |
| | 4 | GND |
|  | C → Connector (male) in module, feed for I/O power supply<br>D → Connection (female) in module, routing of I/O power supply | |

## 8 Pinout

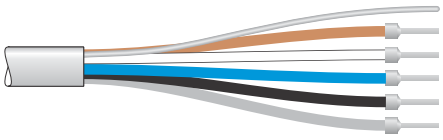



X1 and X3
M12 ①



| Shield | | |
|---|---|---|
| 1 | ABR-B\; | DI/DO |
| 2 | | DI/DO; AB-A |
| 3 | ABR-R; SSI clock; | DI/DO |
| 4 | ABR-R\; SSI clock\; | DI/DO |
| 5 | ABR-A; SSI data; | DI/DO |
| 6 | ABR-A\; SSI data\; | DI/DO |
| 7 | | DI/DO; AB-B |
| 8 | ABR-B; | DI/DO |
| 9 | Reserved | |
| 10 | +5 VDC | |
| 11 | +24 VDC | |
| 12 | GND | |

X2 and X4
M12 ②



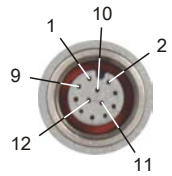
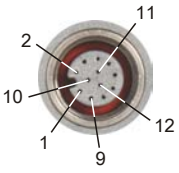
| Shield | |
|---|---|
| 1 | +24 VDC |
| 2 | DI/DO; PWM; AB-A |
| 3 | GND |
| 4 | DI/DO; AB-B |
| 5 | Shield |

① X67CA0I41.xxxx: Straight multi-function cable
X67CA0I51.xxxx: Angled multi-function cable
② X67CA0A41.xxxx: M12 straight sensor cable
X67CA0A51.xxxx: M12 angled sensor cable

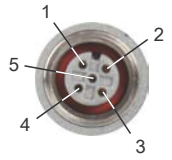
## 8.1 Connector X1 and X3

4 pinouts are listed in the following table:

- I/O channels: Division of digital I/O channels (5 VDC differential signal and 24 VDC)
- ABR incremental encoder: Encoder operated as incremental encoder
- SSI absolute encoder: Encoder operated as SSI absolute encoder
- AB counters: 24 V inputs are operated as AB counters

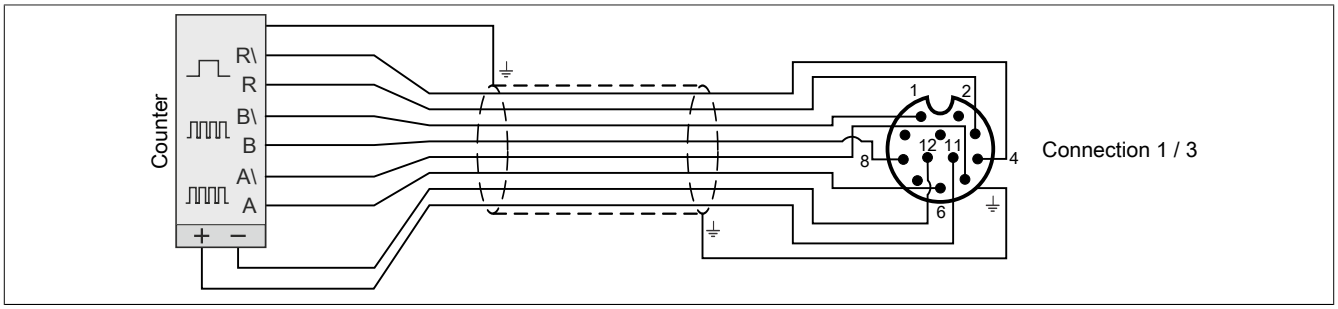
| M12, 12-pin | Pinout | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Connection 1 | Pin | Assignment | Voltage | | I/O channel | | ABR encoders | SSI encoder | AB counter |
| | | | 5 V | 24 V | X1 | X3 | X1: Encoder 1 X3: Encoder 2 | X1: Encoder 1 X3: Encoder 2 | X1: Counter 1 X2: Counter 3 |
|  | 1 | Inputs/Outputs | ● | | 10\ | 14\ | B\ | - | - |
| | 2 | Inputs/Outputs | | ● | 1 | 5 | - | - | A |
| | 3 | Inputs/Outputs | ● | | 11 | 15 | R | Clock | - |
| | 4 | Inputs/Outputs | ● | | 11\ | 15\ | R\ | Clock\ | - |
| | 5 | Inputs/Outputs | ● | | 9 | 13 | A | Data | - |
| | 6 | Inputs/Outputs | ● | | 9\ | 13\ | A\ | Data\ | - |
| | 7 | Inputs/Outputs | | ● | 2 | 6 | - | - | B |
| | 8 | Inputs/Outputs | ● | | 10 | 14 | B | - | - |
| | 9 | Reserved | | | | | | | |
| | 10 | +5 VDC encoder power supply | | | | | | | |
|  | 11 | +24 VDC encoder power supply | | | | | | | |
| Connection 3 | 12 | GND | | | | | | | |

## 8.2 Connection X2 and X4

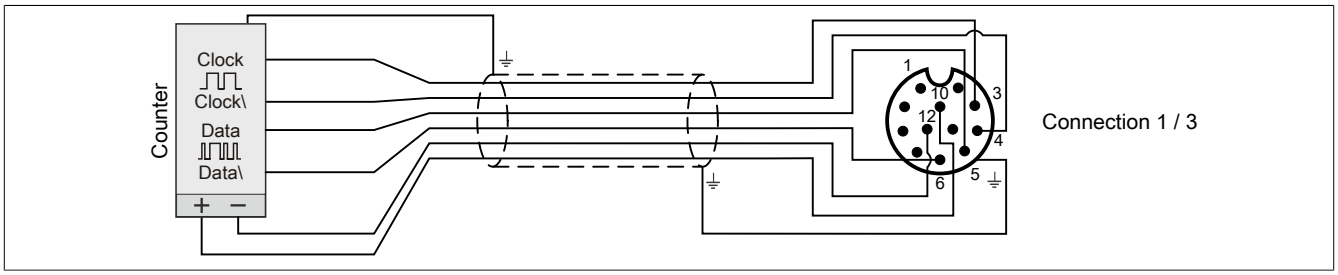
| M12, 5-pin | Pinout | | | | | |
|---|---|---|---|---|---|---|
| Connection 2 | Pin | Assignment | I/O channel | | PWM | | AB counter |
| | | | X2 | X4 | X2 | X4 | X2: Encoder 2 X4: Encoder 4 |
|  | 1 | 24 VDC sensor/actuator power supply[1)] | | | | | |
| | 2 | Input/Output | 3 | 7 | PWM 1 | PWM 2 | A |
| | 3 | GND | | | | | |
| | 4 | Input/Output | 4 | 8 | - | - | B |
|  | 5 | Shield[2)] | | | | | |
| Connection 4 | 1) Sensors/Actuators are not permitted to be supplied externally.<br>2) Shielding also provided by threaded insert in the module.<br><br>X2 and X4 → A-keyed (female), input | | | | | | |

# 9 Connection examples

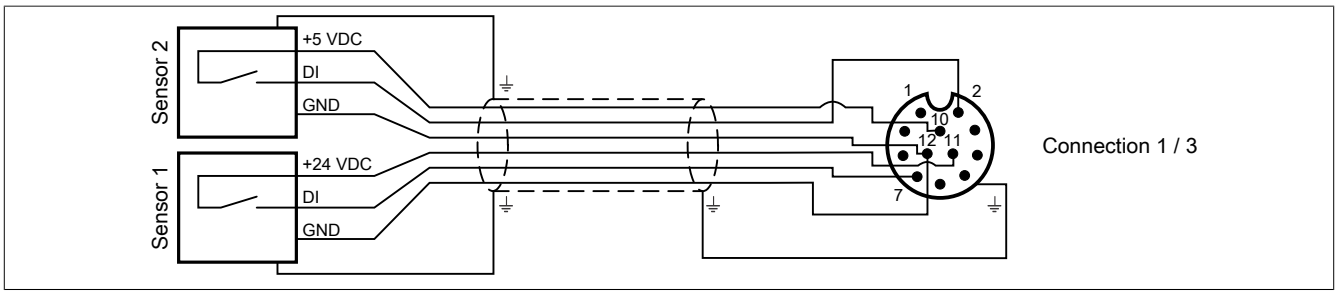
### ABR encoder with 24 V power supply



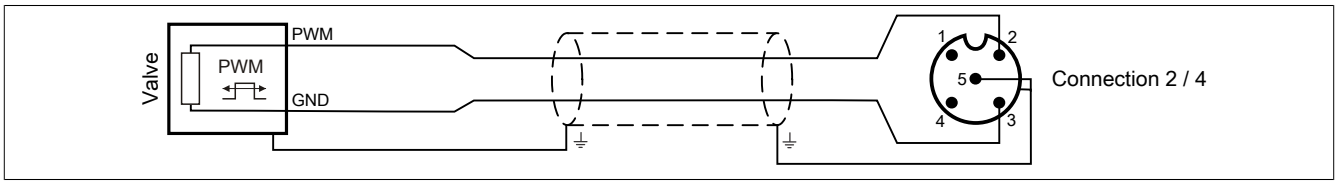

### SSI encoder with 5 V power supply




### Sensor connection with 5 V and 24 V power supply

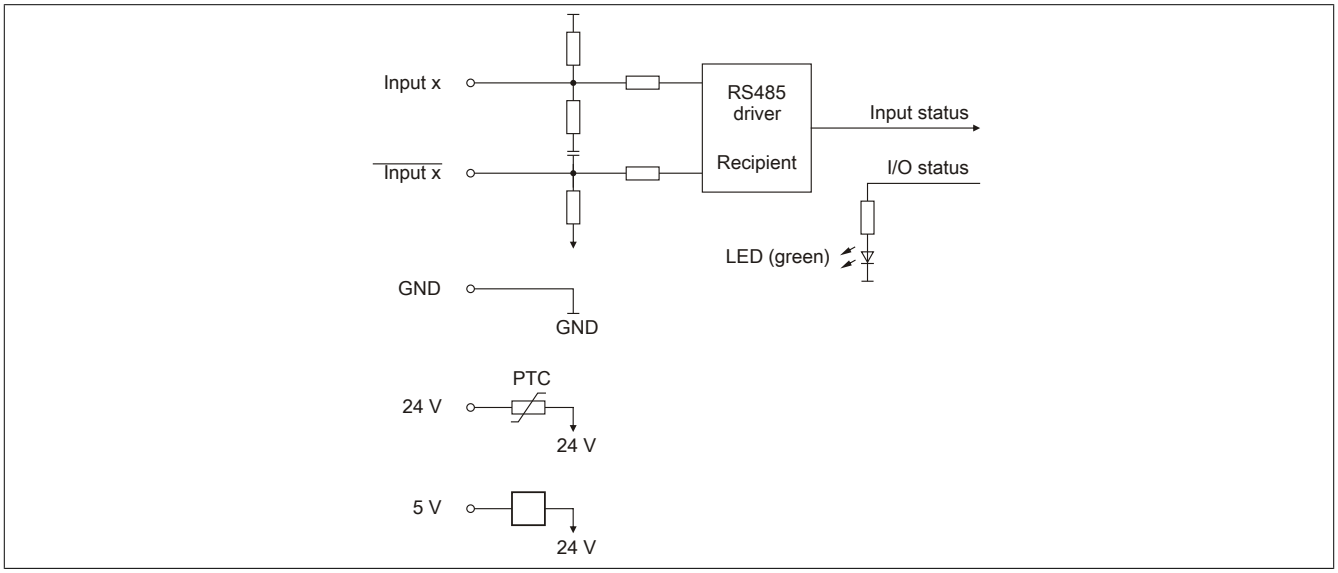



### PWM output

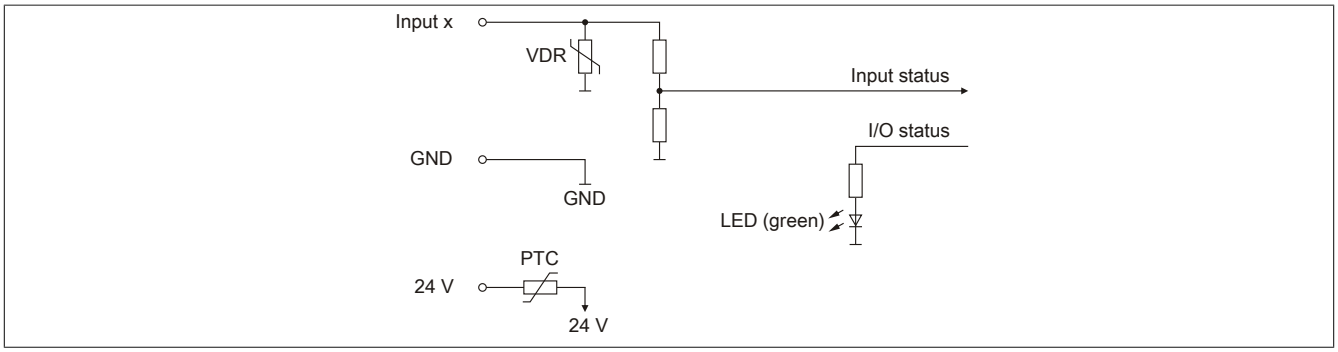


# 10 Input circuit diagram

### 5 V input

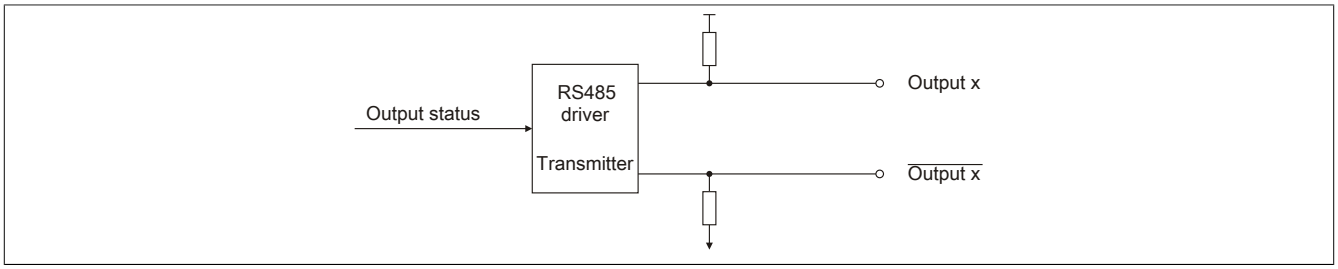



### 24 V input

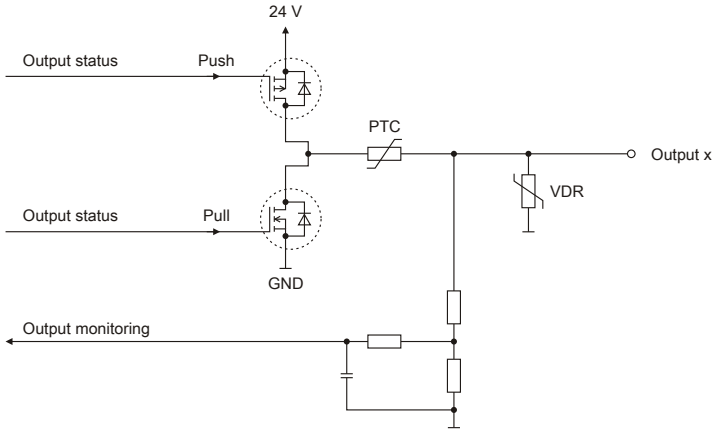



# 11 ABR/SSI - Output circuit diagram

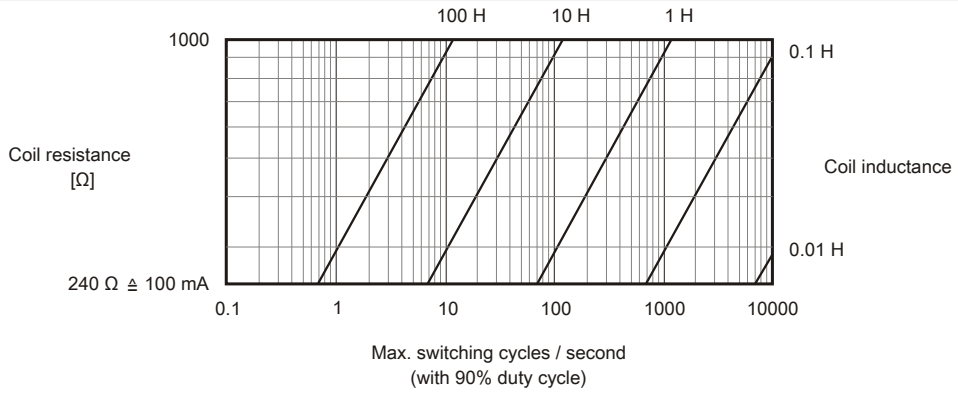
### 5 V output



### 24 V output




## 12 Switching inductive loads



# 13 Description of channel assignments

The functions listed here are directly assigned to the respective hardware channels and cannot be changed:

| Channel | Signal connections |
|---|---|
| 1 | • Digital input/output 1 (24 VDC)<br>• Event counter 1<br>• AB counter 1, signal line A<br>• Up/down counter 1 - frequency |
| 2 | • Digital input/output 2 (24 VDC)<br>• Event counter 2<br>• AB counter 1, signal line B<br>• Up/down counter 1 - direction |
| 3 | • Digital input/output 3 (24 VDC)<br>• Event counter 3<br>• AB counter 2, signal line A<br>• Up/down counter 2 - frequency<br>• PWM output 1 |
| 4 | • Digital input/output 4 (24 VDC)<br>• Event counter 4<br>• AB counter 2, signal line B<br>• Up/down counter 2 - direction |
| 5 | • Digital input/output 5 (24 VDC)<br>• Event counter 5<br>• AB counter 3, signal line A<br>• Up/down counter 3 - frequency |
| 6 | • Digital input/output 6 (24 VDC)<br>• Event counter 6<br>• AB counter 3, signal line B<br>• Up/down counter 3 - direction |
| 7 | • Digital input/output 7 (24 VDC)<br>• Event counter 7<br>• AB counter 4, signal line A<br>• Up/down counter 4 - frequency<br>• PWM output 2 |
| 8 | • Digital input/output 8 (24 VDC)<br>• Event counter 8<br>• AB counter 4 - signal line B<br>• Up/down counter 4 - direction |
| 9 | • Digital input/output 9 (5 VDC)<br>• Event counter 9<br>• ABR counter 1, signal line A<br>• SSI encoder 1 - data line |
| 10 | • Digital input/output 10 (5 VDC)<br>• Event counter 10<br>• ABR encoder 1 - signal line B |
| 11 | • Digital input/output 11 (5 VDC)<br>• Event counter 11<br>• ABR encoder 1 - signal line R<br>• SSI encoder 1 - clock line |
| 12 | Not used |
| 13 | • Digital input/output 13 (5 VDC)<br>• Event counter 13<br>• ABR encoder 2 - signal line A<br>• SSI encoder 2 - data line |
| 14 | • Digital input/output 14 (5 VDC)<br>• Event counter 14<br>• ABR encoder 2 - signal line B |
| 15 | • Digital input/output 15 (5 VDC)<br>• Event counter 15<br>• ABR encoder 2 - signal line R<br>• SSI encoder 2 - clock line |

Options available in addition to these basic functions, such as comparator outputs or latch inputs, can be configured freely to unused input/output channels.

# 14 Register description

## 14.1 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 X67 system user's manual.

## 14.2 Function model 0 "16-bit counter" and function model 1 "32-bit counter"

The following 2 models can be selected:

- 16-bit counter, Function model 0
- 32-bit counter, Function model 1 (identified in the table with a "(D)" in the data type and "(_32Bit)" in the name.)

The only difference between these two models is that they use either 16-bit or 32-bit registers for incremental counter functions. The following belong to this group:

- ABR encoders
- AB counters
- Up/down counters
- Event counter

All other module functions e.g. SSI, PWM and time measurement, as well as their data types, are identical for the two models.

| Register | Name | Data type | Read | | Write | |
|---|---|---|---|---|---|---|
| | | | Cyclic | Acyclic | Cyclic | Acyclic |
| **Module configuration - General** | | | | | | |
| (N-1) * 2 | CfO_CFGchannelN (index N = 01 to 15) | USINT | | | | • |
| 64 + N * 2 | CfO_LEDNsource (index N = 0 to 7) | USINT | | | | • |
| **Configuration - Input for ABR encoders** | | | | | | |
| 512 | CfO_DIREKTIOevent0IDwr | UINT | | | | • |
| 544 | CfO_DIREKTIOevent1IDwr | UINT | | | | • |
| 516 | CfO_DIREKTIOevent0mode | USINT | | | | • |
| 548 | CfO_DIREKTIOevent1mode | USINT | | | | • |
| 522 | CfO_DIREKTIOevent0compState | UINT | | | | • |
| 554 | CfO_DIREKTIOevent1compState | UINT | | | | • |
| 512 | CfO_DIREKTIOevent0compState | UINT | | | | • |
| 544 | CfO_DIREKTIOevent1compState | UINT | | | | • |
| 520 | CfO_Ev0CompMask | UINT | | | | • |
| 552 | CfO_Ev1CompMask | UINT | | | | • |
| 3088 | CfO_Counter5PresetValue1(_32Bit) | U(D)INT | | | | • |
| 3600 | CfO_Counter7PresetValue1(_32Bit) | U(D)INT | | | | • |
| 3092 | CfO_Counter5PresetValue2(_32Bit) | U(D)INT | | | | • |
| 3604 | CfO_Counter7PresetValue2(_32Bit) | U(D)INT | | | | • |
| 3072 | CfO_Counter5config | USINT | | | | • |
| 3584 | CfO_Counter7config | USINT | | | | • |
| 3080 | CfO_Counter5configReg0 | USINT | | | | • |
| 3592 | CfO_Counter7configReg0 | USINT | | | | • |
| 3082 | CfO_Counter5configReg1 | USINT | | | | • |
| 3594 | CfO_Counter7configReg1 | USINT | | | | • |
| 3136 | CfO_Counter5event0IDwr | UINT | | | | • |
| 3648 | CfO_Counter7event0IDwr | UINT | | | | • |
| 3168 | CfO_Counter5event1IDwr | UINT | | | | • |
| 3680 | CfO_Counter7event1IDwr | UINT | | | | • |
| 3144 | CfO_Counter5event0config | UINT | | | | • |
| 3656 | CfO_Counter7event0config | UINT | | | | • |
| 3176 | CfO_Counter5event1config | UINT | | | | • |
| 3688 | CfO_Counter7event1config | UINT | | | | • |
| 3172 | CfO_Counter5event1mode | USINT | | | | • |
| 3684 | CfO_Counter7event1mode | USINT | | | | • |
| **Configuration - Inputs for event counters** | | | | | | |
| 2056 + (N-1) * 256 | CfO_CounterNconfigReg0 (index N = 1 to 8) | USINT | | | | • |
| 2058 + (N-1) * 256 | CfO_CounterNconfigReg1 (index N = 1 to 8)[1)] | USINT | | | | • |
| 2112 + (N-1) * 256 | CfO_CounterNevent0IDwr (index N = 1 to 8) | UINT | | | | • |
| 2120 + (N-1) * 256 | CfO_CounterNevent0config (index N = 1 to 8) | UINT | | | | • |
| 2116 + (N-1) * 256 | CfO_CounterNevent0mode (index N = 1 to 8) | USINT | | | | • |

| Register | Name | Data type | Read | | Write | |
|---|---|---|---|---|---|---|
| | | | Cyclic | Acyclic | Cyclic | Acyclic |
| 2144 + (N-1) * 256 | CfO_CounterNevent1IDwr (index N = 1 to 8)[1] | UINT | | | | ● |
| 2152 + (N-1) * 256 | CfO_CounterNevent1config (index N = 1 to 8)[1] | UINT | | | | ● |
| 2148 + (N-1) * 256 | CfO_CounterNevent1mode (index N = 1 to 8)[1] | USINT | | | | ● |
| **Configuration - Inputs for AB and up/down counters** | | | | | | |
| 2048 + (N-1) * 256 | CfO_CounterNconfig (index N = 1 to 4) | USINT | | | | ● |
| 2056 + (N-1) * 256 | CfO_CounterNconfigReg0 (index N = 1 to 4) | USINT | | | | ● |
| 2058 + (N-1) * 256 | CfO_CounterNconfigReg1 (index N = 1 to 4) | USINT | | | | ● |
| 2112 + (N-1) * 256 | CfO_CounterNevent0IDwr (index N = 1 to 4) | UINT | | | | ● |
| 2120 + (N-1) * 256 | CfO_CounterNevent0config (index N = 1 to 4) | UINT | | | | ● |
| 2116 + (N-1) * 256 | CfO_CounterNevent0mode (index N = 1 to 4) | USINT | | | | ● |
| 2144 + (N-1) * 256 | CfO_CounterNevent1IDwr (index N = 1 to 4) | UINT | | | | ● |
| 2152 + (N-1) * 256 | CfO_CounterNevent1config (index N = 1 to 4) | UINT | | | | ● |
| 2148 + (N-1) * 256 | CfO_CounterNevent1mode (index N = 1 to 4) | USINT | | | | ● |
| **Configuration - Inputs for SSI encoders** | | | | | | |
| 7176 | CfO_SSI1cfg | UINT | | | | ● |
| 7432 | CfO_SSI2cfg | UINT | | | | ● |
| 7180 | CfO_SSI1control | USINT | | | | ● |
| 7436 | CfO_SSI2control | USINT | | | | ● |
| 7168 | CfO_SSI1eventIDwr | UINT | | | | ● |
| 7424 | CfO_SSI2eventIDwr | UINT | | | | ● |
| 7232 | CfO_SSI1event0IDwr | UINT | | | | ● |
| 7488 | CfO_SSI2event0IDwr | UINT | | | | ● |
| 7240 | CfO_SSI1event0config | UINT | | | | ● |
| 7496 | CfO_SSI2event0config | UINT | | | | ● |
| 7236 | CfO_SSI1event0mode | USINT | | | | ● |
| 7492 | CfO_SSI2event0mode | USINT | | | | ● |
| 7172 | ConfigAdvanced01 | UDINT | | | | ● |
| 7428 | ConfigAdvanced02 | UDINT | | | | ● |
| **Configuration - Comparator function for ABR and SSI encoders as well as AB and up/down counters** | | | | | | |
| 256 | CfO_OutClearMask | UINT | | | | ● |
| 258 | CfO_OutSetMask | UINT | | | | ● |
| 1024 + N * 32 | CfO_DIREKTIOouteventNIDwr (index N = 0 to 3) | UINT | | | | ● |
| 1034 + N * 32 | CfO_DIREKTIOoutsetmaskN (index N = 0 to 3) | UINT | | | | ● |
| 1032 + N * 32 | CfO_DIREKTIOoutclearmaskN (index N = 0 to 3) | UINT | | | | ● |
| **Configuration - Outputs for PWM (pulse width modulation)** | | | | | | |
| 6144 | CfO_PWM0prescaler | UINT | | | | ● |
| 6160 | CfO_PWM1prescaler | UINT | | | | ● |
| **Module communication - General** | | | | | | |
| 40 | Status of encoder supplies | USINT | ● | | | |
| | PowerSupply01 | Bit 0 | | | | |
| | PowerSupply02 | Bit 1 | | | | |
| **Communication - Digital inputs** | | | | | | |
| 264 | DigitalInput1_16 | UINT | ● | | | |
| | DigitalInput01 | Bit 0 | | | | |
| | ... | ... | | | | |
| | DigitalInput15 | Bit 14 | | | | |
| **Communication - Digital outputs** | | | | | | |
| 260 | DigitalOutput1_16 | UINT | | | ● | |
| | DigitalOutput01 | Bit 0 | | | | |
| | ... | ... | | | | |
| | DigitalOutput15 | Bit 14 | | | | |
| 264 | Status of the digital outputs | UINT | ● | | | |
| | StatusDigitalOutput01 | Bit 0 | | | | |
| | ... | ... | | | | |
| | StatusDigitalOutput15 | Bit 14 | | | | |
| **Communication - Event counters** | | | | | | |
| 2080 + (N-1) * 256 | EventCounter01 (index N = 1, 3, 5 ... 15) | U(D)INT | ● | | | |
| 2084 + (N-1) * 256 | EventCounter02 (index N = 2, 4, 6 ... 14)[2] | U(D)INT | ● | | | |
| **Communication - Input for ABR encoders (optionally with comparator)** | | | | | | |
| 3104 | ABRConnector01 | (D)INT | ● | | | |
| 3616 | ABRConnector03 | (D)INT | ● | | | |
| 3140 | ReferenceModeABRConnector01 | USINT | | | ● | |
| 3652 | ReferenceModeABRConnector03 | USINT | | | ● | |
| 3184 | OriginComparatorABRConnector01 | (D)INT | | | ● | |
| 3696 | OriginComparatorABRConnector03 | (D)INT | | | ● | |
| 3188 | MarginComparatorABRConnector01 | U(D)INT | | | ● | |
| 3700 | MarginComparatorABRConnector03 | U(D)INT | | | ● | |
| 264 | Input states of the channels | UINT | ● | | | |
| | ReferenceEnableSwitchABRConnector01 or ReferenceEnableSwitchABRConnector03 (without comparator) ComparatorActualValueABRConnector01 or ComparatorActualValueABRConnecto03 (with comparator) | Bit x | | | | |
| 3196 | Latch01ABR01 | (D)INT | ● | | | |

| Register | Name | Data type | Read | | Write | |
|---|---|---|---|---|---|---|
| | | | Cyclic | Acyclic | Cyclic | Acyclic |
| 3708 | Latch01ABR02 | (D)INT | • | | | |
| 3142 | StatusABRConnector01 | USINT | • | | | |
| 3654 | StatusABRConnector03 | USINT | • | | | |
| **Communication - Input for AB counters** | | | | | | |
| 2080 + (N-1) * 256 | ABConnector0N (index N = 1 to 4) | (D)INT | • | | | |
| 2160 + (N-1) * 256 | OriginComparatorABConnector0N (index N = 1 to 4) | (D)INT | | | • | |
| 2164 + (N-1) * 256 | MarginComparatorABConnector01N (index N = 1 to 4) | U(D)INT | | | • | |
| 264 | Input states of the channels | UINT | • | | | |
| | ComparatorActualValueCounterConnector0N (index N = 1 to 4) | Bit x | | | | |
| 2140 + (N-1) * 256 | Latch01AB0N (index N = 1 to 4) | (D)INT | • | | | |
| 2172 + (N-1) * 256 | Latch02AB0N (index N = 1 to 4) | (D)INT | • | | | |
| **Communication - Up/Down counters** | | | | | | |
| 2080 + (N-1) * 256 | CounterConnector0N (index N = 1 to 4) | U(D)INT | • | | | |
| 2160 + (N-1) * 256 | OriginComparatorCounterConnector0N (index N = 1 to 4) | U(D)INT | | | • | |
| 2164 + (N-1) * 256 | MarginComparatorCounterConnector0N (index N = 1 to 4) | U(D)INT | | | • | |
| 264 | Input states of the channels | UINT | • | | | |
| | ComparatorActualValueCounterConnector0N (index N = 1 to 4) | Bit x | | | | |
| 2140 + (N-1) * 256 | Latch01CounterConnector0N (index N = 1 to 4) | U(D)INT | • | | | |
| 2172 + (N-1) * 256 | Latch02CounterConnector0N (index N = 1 to 4) | U(D)INT | • | | | |
| **Communication - Input for SSI encoders** | | | | | | |
| 7184 | SSIConnector01 | UDINT | • | | | |
| 7440 | SSIConnector03 | UDINT | • | | | |
| 3108 | EventCounter10 | UINT | • | | | |
| 3620 | EventCounter14 | UINT | • | | | |
| 7248 | OriginComparatorSSIConnector01 | UDINT | | | • | |
| 7504 | OriginComparatorSSIConnector03 | UDINT | | | • | |
| 7252 | MarginComparatorSSIConnector01 | UDINT | | | • | |
| 7508 | MarginComparatorSSIConnector03 | UDINT | | | • | |
| 264 | Input states of the channels | UINT | • | | | |
| | ComparatorActualValueSSIConnector01 or ComparatorActualValueSSIConnector01 | Bit x | | | | |
| 7260 | Latch01SSIConnector01 | UDINT | • | | | |
| 7516 | Latch01SSIConnector03 | UDINT | • | | | |
| **Communication - Outputs for PWM (pulse width modulation)** | | | | | | |
| 6146 | PWMOutput03 | UINT | | | • | |
| 6162 | PWMOutput07 | UINT | | | • | |
| **Configuration - Edge detection** | | | | | | |
| 4104 | CfO_EdgeDetectFalling | UINT | | | | • |
| 4106 | CfO_EdgeDetectRising | UINT | | | | • |
| 4108 | CfO_FallingDisProtection | UINT | | | | • |
| 4110 | CfO_RisingDisProtection | UINT | | | | • |
| **Configuration - Time measurement** | | | | | | |
| 4336 | CfO_EdgeTimeglobalenable | USINT | | | | • |
| 4344 + N * 8[2)] | CfO_EdgeTimeFallingModeN (index N = 01 to 15) | UINT | | | | • |
| 4472 + N * 8[2)] | CfO_EdgeTimeRisingModeN (index N = 01 to 15) | UINT | | | | • |
| **Communication - Time measurement** | | | | | | |
| 4342 | Trigger rising edge detection | USINT | | | • | |
| | TriggerRisingCH01 | Bit 0 | | | | |
| | ... | ... | | | | |
| | TriggerRisingCH08 | Bit 7 | | | | |
| 4343 | Trigger rising edge detection | USINT | | | • | |
| | TriggerRisingCH09 | Bit 0 | | | | |
| | ... | ... | | | | |
| | TriggerRisingCH15 | Bit 6 | | | | |
| 4350 | Show first rising trigger edge | USINT | • | | | |
| | BusyTriggerRisingCH01 | Bit 0 | | | | |
| | ... | ... | | | | |
| | BusyTriggerRisingCH08 | Bit 7 | | | | |
| 4351 | Show first rising trigger edge | USINT | • | | | |
| | BusyTriggerRisingCH09 | Bit 0 | | | | |
| | ... | ... | | | | |
| | BusyTriggerRisingCH15 | Bit 6 | | | | |
| 4340 | Trigger falling edge detection | USINT | | | • | |
| | TriggerFallingCH01 | Bit 0 | | | | |
| | ... | ... | | | | |
| | TriggerFallingCH08 | Bit 7 | | | | |
| 4341 | Trigger falling edge detection | USINT | | | • | |
| | TriggerFallingCH09 | Bit 0 | | | | |
| | ... | ... | | | | |
| | TriggerFallingCH15 | Bit 6 | | | | |
| 4348 | Show first falling trigger edge | USINT | • | | | |
| | BusyTriggerFallingCH01 | Bit 0 | | | | |

| Register | Name | Data type | Read | | Write | |
|---|---|---|---|---|---|---|
| | | | Cyclic | Acyclic | Cyclic | Acyclic |
| | ... | ... | | | | |
| | BusyTriggerFallingCH08 | Bit 7 | | | | |
| 4349 | Show first falling trigger edge | USINT | • | | | |
| | BusyTriggerFallingCH09 | Bit 0 | | | | |
| | ... | ... | | | | |
| | BusyTriggerFallingCH15 | Bit 6 | | | | |
| 4474 + N * 8[2)] | CountRisingCHN (index N = 01 to 15) | USINT | • | | | |
| 4476 + N * 8[2)] | TimeStampRisingCHN (index N = 01 to 15) | UINT | • | | | |
| 4478 + N * 8[2)] | TimeDiffRisingCHN (index N = 01 to 15) | UINT | • | | | |
| 4346 + N * 8[2)] | CountFallingCHN (index N = 01 to 15) | USINT | • | | | |
| 4348 + N * 8[2)] | TimeStampFallingCHN (index N = 01 to 15) | UINT | • | | | |
| 4350 + N * 8[2)] | TimeDiffFallingCHN (index N = 01 to 15) | UINT | • | | | |

1) Registers with index value 6 and 8 are not used.
2) Register with index value 12 does not exist.

## 14.3 Function model 254 - Bus controller

Unlike the function models 0 and 1, this model only offers a selection of functions with a limited scope of configuration on the module.

The following functions are provided and can be run at the same time:

- 1 SSI encoder
- 1 ABR encoder with configurable reference pulse edge and reference position
- 1 event counter with configurable counting direction
- 3 AB counters
- 1 PWM output

| Register | Offset[1)] | Name | Data type | Read | | Write | |
|---|---|---|---|---|---|---|---|
| | | | | Cyclic | Acyclic | Cyclic | Acyclic |
| **Configuration - Event counter** | | | | | | | |
| 2816 | - | CfO_Counter4config | USINT | | | | • |
| 2824 | - | CfO_Counter4configReg0 | USINT | | | | • |
| 2826 | - | CfO_Counter4configReg1 | USINT | | | | • |
| **Configuration - ABR encoder** | | | | | | | |
| 3088 | - | CfO_Counter5PresetValue1 | UINT | | | | • |
| 3092 | - | CfO_Counter5PresetValue2 | UINT | | | | • |
| 3072 | - | CfO_Counter5config | USINT | | | | • |
| 3080 | - | CfO_Counter5configReg0 | USINT | | | | • |
| 3082 | - | CfO_Counter5configReg1 | USINT | | | | • |
| 3136 | - | CfO_Counter5event0IDwr | UINT | | | | • |
| 3144 | - | CfO_Counter5event0config | UINT | | | | • |
| 512 | - | CfO_DIREKTIOevent0IDwr | UINT | | | | • |
| 516 | - | CfO_DIREKTIOevent0mode | USINT | | | | • |
| **Configuration - AB counter** | | | | | | | |
| 2048 + (N-1) * 256 | - | CfO_CounterNconfig (index N = 1 to 3) | USINT | | | | • |
| 2056 + (N-1) * 256 | - | CfO_CounterNconfigReg0 (index N = 1 to 3) | USINT | | | | • |
| 2058 + (N-1) * 256 | - | CfO_CounterNconfigReg1 (index N = 1 to 3) | USINT | | | | • |
| **Configuration - Inputs for SSI encoders** | | | | | | | |
| 7424 | - | CfO_SSI2eventIDwr | UINT | | | | • |
| 7428 | - | ConfigAdvanced02 | UDINT | | | | • |
| **Configuration - Outputs for PWM (pulse width modulation)** | | | | | | | |
| 6160 | - | CfO_PWM1prescaler | UINT | | | | • |
| **Module communication - General** | | | | | | | |
| 40 | 3 | Status of encoder supplies | USINT | • | | | |
| | | PowerSupply01 | Bit 0 | | | | |
| | | PowerSupply02 | Bit 1 | | | | |
| **Communication - Event counters** | | | | | | | |
| 2852 | 14 | EventCounter08 | UINT | • | | | |
| **Communication - Input for ABR encoders** | | | | | | | |
| 3104 | 0 | ABRConnector01 | INT | • | | | |
| 3140 | 0 | ReferenceModeABRConnector01 | USINT | | | • | |
| 3142 | 2 | StatusABRConnector01 | USINT | • | | | |
| **Communication - Input for AB counters** | | | | | | | |
| 2080 | 8 | ABConnector01 | INT | • | | | |
| 2336 | 10 | ABConnector02 | INT | • | | | |
| 2592 | 12 | ABConnector02 | INT | • | | | |

| Register | Offset[1)] | Name | Data type | Read | | Write | |
|---|---|---|---|---|---|---|---|
| | | | | Cyclic | Acyclic | Cyclic | Acyclic |
| **Communication - Input for SSI encoders** | | | | | | | |
| 7440 | 4 | SSIConnector03 | UDINT | • | | | |
| **Communication - Outputs for PWM (pulse width modulation)** | | | | | | | |
| 6162 | 2 | PWMOutput07 | UINT | | | • | |

1) The offset specifies the position of the register within the CAN object.

### 14.3.1 Using the module on the bus controller

Function model 254 "Bus controller" is used by default only by non-configurable bus controllers. All other bus controllers can use additional registers and functions depending on the fieldbus used.

For detailed information, see section "Additional information - Using I/O modules on the bus controller" of the X67 user's manual (version 3.30 or later).

### 14.3.2 CAN I/O bus controller

The module occupies 2 analog logical slots on CAN I/O.

## 14.4 General module registers

### 14.4.1 Configuring LED status indicators

Name:
CfO_LED0source to CfO_LED7source

These registers can be used to define how the module's LED status indicators are used. Blinking patterns can be generated from the application, and the status of the physical inputs and outputs can be indicated.

The following applies:

| | Connection | LED |
|---|---|---|
| CfO_LED0source | 1 | Green |
| CfO_LED1source | 1 | Orange |
| ... | ... | ... |
| CfO_LED6source | 4 | Green |
| CfO_LED7source | 4 | Orange |

| Data type | Values |
|---|---|
| USINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 - 3 | MODE = 0 | 0 | LED off |
| | | 1 | Blinking quickly |
| | | 2 | Blinking |
| | | 3 | Blinking slowly |
| | | 4 | Single flash |
| | | 5 | Double flash |
| | | 6 to 15 | Reserved |
| | MODE = 1 (inverted) | 0 | LED on |
| | | 1 | Blinking quickly |
| | | 2 | Blinking |
| | | 3 | Blinking slowly |
| | | 4 | Single flash |
| | | 5 | Double flash |
| | | 6 to 15 | Reserved |
| | MODE = 2 | 0 to 15 | Number of the physical input channel |
| | MODE = 3 | 0 to 15 | Number of the physical output channel |
| 4 - 7 | Selection of the mode for the LED status indicator | 0 | LED blinking pattern |
| | | 1 | Inverted LED blinking pattern |
| | | 2 | Displays a channel's physical input status |
| | | 3 | Displays a channel's physical output status |
| | | 4 to 15 | Reserved |

### 14.4.2 Status of encoder supplies

Name:
PowerSupply01 to PowerSupply02

This register shows the status of the integrated encoder supplies. A faulty encoder power supply is displayed as a warning.

| Data type | Value |
|---|---|
| USINT | See bit structure. |

Bit structure:

| Bit | Name | Value | Information |
|---|---|---|---|
| 0 | PowerSupply01 | 0 | 24 VDC encoder power supply OK |
| | | 1 | 24 VDC encoder power supply faulty |
| 1 | PowerSupply02 | 0 | 5 VDC encoder power supply OK |
| | | 1 | 5 VDC encoder power supply faulty |
| 2 - 7 | Reserved | - | |

## 14.5 Digital inputs and outputs

### 14.5.1 Configure physical channels

Name:
CfO_CFGchannel01 to CfO_CFGchannel15

This register can be used to configure physical I/O channels 1 to 15.

> **Information:**
>
> **CfO_CFGchannel12 is not connected to a physical I/O channel.**

| Data type | Values |
|---|---|
| USINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | Pull[1)] | 0 | Disabled |
| | | 1 | Enabled |
| 1 | Push[1)] | 0 | Disabled |
| | | 1 | Enabled |
| 2 | Inverted input | 0 | Disabled |
| | | 1 | Enabled |
| 3 | Inverted output | 0 | Disabled |
| | | 1 | Enabled |
| 4 - 7 | Output type | 0 | Direct I/O |
| | | 1 to 5 | Reserved |
| | | 6 | PWM (channel-specific) |
| | | 7 | Reserved |

1) To configure a channel as an output, Push and/or Pull must be enabled.

### 14.5.2 Reset mask of the digital channels

Name:
CfO_OutClearMask

The settings in this register only affect the values written to register "DigitalOutput xx" on page 19.

- 0 allows manual reset of digital outputs using registers DigitalOutput01 to 15.
- 1 prevents manual reset of digital outputs using registers DigitalOutput01 to 15.

When "1" is used, the output event function can be used to reset the outputs.

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | DigitalOutput01 | 0 | Writing 0 to register DigitalOutput01 resets the output. |
| | | 1 | Writing 0 to register DigitalOutput01 does not reset the output. |
| ... | ... | ... | |
| 10 | DigitalOutput11 | 0 | Writing 0 to register DigitalOutput11 resets the output. |
| | | 1 | Writing 0 to register DigitalOutput11 does not reset the output. |
| 11 | Reserved (output 12 does not exist) | - | |
| 12 | DigitalOutput13 | 0 | Writing 0 to register DigitalOutput13 resets the output. |
| | | 1 | Writing 0 to register DigitalOutput13 does not reset the output. |
| ... | ... | ... | |
| 14 | DigitalOutput15 | 0 | Writing 0 to register DigitalOutput15 resets the output. |
| | | 1 | Writing 0 to register DigitalOutput15 does not reset the output. |
| 15 | Reserved | - | |

### 14.5.3 Set mask of the digital channels

Name:
CfO_OutSetMask

The settings in this register only affect the values written to register "DigitalOutput xx" on page 19.

- 0 allows manual setting of digital outputs using registers DigitalOutput01 to 15.
- 1 prevents manual setting of digital outputs using registers DigitalOutput01 to 15.

When "1" is used, the output event function can be used to reset the outputs.

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | DigitalOutput01 | 0 | Writing 1 to register DigitalOutput01 sets the output. |
| | | 1 | Writing 1 to register DigitalOutput01 does not set the output. |
| ... | ... | ... | |
| 10 | DigitalOutput11 | 0 | Writing 1 to register DigitalOutput11 sets the output. |
| | | 1 | Writing 1 to register DigitalOutput11 does not set the output. |
| 11 | Reserved (output 12 does not exist) | - | |
| 12 | DigitalOutput13 | 0 | Writing 1 to register DigitalOutput13 sets the output. |
| | | 1 | Writing 1 to register DigitalOutput13 does not set the output. |
| ... | ... | ... | |
| 14 | DigitalOutput15 | 0 | Writing 1 to register DigitalOutput15 sets the output. |
| | | 1 | Writing 1 to register DigitalOutput15 does not set the output. |
| 15 | Reserved | - | |

### 14.5.4 Input states of the digital inputs

Name:
DigitalInput1_16
DigitalInput01 to DigitalInput11
DigitalInput13 to DigitalInput15

This register reads the input status of a physical channel. The polarity settings are accounted for in the value (bit 2 in "CfO_CFGchannel[x]" on page 17 register).

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Name | Value | Information |
|---|---|---|---|
| 0 | DigitalInput01 | 0 or 1 | Input state of the physical channel |
| .. | ... | ... | |
| 10 | DigitalInput11 | 0 or 1 | Input state of the physical channel |
| 11 | Reserved (input 12 does not exist) | - | |
| 12 | DigitalInput13 | 0 or 1 | Input state of the physical channel |
| ... | ... | ... | |
| 14 | DigitalInput15 | 0 or 1 | Input state of the physical channel |

### 14.5.5 Output states of the channels

Name:
DigitalOutput1_16
DigitalOutput01 to DigitalOutput11
DigitalOutput13 to DigitalOutput15

The output status of a physical channel can be written using this register. In order to configure a channel as an output:

1) Bit 0 "Push" and/or bit 1 "Pull" must be enabled in the "CfO_CFGchannel[x]" on page 17 register.
2) Bits 4 to 7 in the "CfO_CFGchannel[x]" on page 17 register must be set to Direct I/O.
3) 0 must be set for the respective channel in the "CfO_OutClearMask" on page 18 and "CfO_OutSetMask" on page 18 registers.

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Name | Value | Information |
|---|---|---|---|
| 0 | DigitalOutput01 | 0 or 1 | Output state of the physical channel |
| .. | ... | ... | |
| 10 | DigitalOutput11 | 0 or 1 | Output state of the physical channel |
| 11 | Reserved (output 12 does not exist) | - | |
| 12 | DigitalOutput13 | 0 or 1 | Output state of the physical channel |
| ... | ... | ... | |
| 14 | DigitalOutput15 | 0 or 1 | Output state of the physical channel |

### 14.5.6 Status of the digital outputs

Name:
StatusDigitalOutput01 to StatusDigitalOutput11
StatusDigitalOutput13 to StatusDigitalOutput15

The state of the output channel passed to the hardware is indicated in this register.

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Name | Value | Information |
|---|---|---|---|
| 0 | StatusDigitalOutput01 | 0 or 1 | State of the hardware output channel |
| .. | ... | ... | |
| 10 | StatusDigitalOutput11 | 0 or 1 | State of the hardware output channel |
| 11 | Reserved (output 12 does not exist) | - | |
| 12 | StatusDigitalOutput13 | 0 or 1 | State of the hardware output channel |
| ... | ... | ... | |
| 14 | StatusDigitalOutput15 | 0 or 1 | State of the hardware output channel |

### 14.5.7 Input states of the channels

Name:
ComparatorActualValueABConnector01 to ComparatorActualValueABConnector04
ComparatorActualValueCounterConnector01 to ComparatorActualValueCounterConnector04
ComparatorActualValueABRConnector01 and ComparatorActualValueABRConnector03
ReferenceEnableSwitchABRConnector01 and ReferenceEnableSwitchABRConnector03
ComparatorActualValueSSIConnector01 and ComparatorActualValueSSIConnector03

This register reads out the input state of a physical channel. The polarity settings are accounted for in the value (bit 2 in "CfO_CFGchannel[x]" on page 17 register).

To improve readability, the bits in this register are shown in the Automation Studio I/O mapping table under different names based on the function being used.

## Information:

**In Automation Studio, only the BOOL value of the channel configured in Automation Studio is returned for each function instead of the entire UINT value.**

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Name | Value | Information |
|---|---|---|---|
| 0 | Channel 1 | 0 or 1 | Input state of the physical channel |
| .. | | | |
| 14 | Channel 15 | 0 or 1 | Input state of the physical channel |

The following table shows which input channels can be assigned to the individual names/functions.

| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 12 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Channel XX** | **01** | **02** | **03** | **04** | **05** | **06** | **07** | **08** | **09** | **10** | **11** | **12** | **13** | **14** | **15** |
| ComparatorActualValueABConnector01 | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | | ● | ● | ● |
| ComparatorActualValueABConnector02 | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | | ● | ● | ● |
| ComparatorActualValueABConnector03 | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | | ● | ● | ● |
| ComparatorActualValueABConnector04 | ● | ● | ● | ● | ● | ● | | | ● | ● | ● | | ● | ● | ● |
| | | | | | | | | | | | | | | | |
| ComparatorActualValueABRConnector01 | ● | ● | ● | ● | ● | ● | ● | ● | | | | | ● | ● | ● |
| ComparatorActualValueABRConnector03 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | |
| | | | | | | | | | | | | | | | |
| ReferenceEnableSwitchABRConnector01 | ● | ● | ● | ● | ● | ● | ● | ● | | | | | ● | ● | ● |
| ReferenceEnableSwitchABRConnector03 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | | |
| | | | | | | | | | | | | | | | |
| ComparatorActualValueCounterConnector01 | | | ● | ● | ● | ● | ● | ● | ● | ● | ● | | ● | ● | ● |
| ComparatorActualValueCounterConnector02 | ● | ● | | | ● | ● | ● | ● | ● | ● | ● | | ● | ● | ● |
| ComparatorActualValueCounterConnector03 | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | | ● | ● | ● |
| ComparatorActualValueCounterConnector04 | ● | ● | ● | ● | ● | ● | | | ● | ● | ● | | ● | ● | ● |
| | | | | | | | | | | | | | | | |
| ComparatorActualValueSSIConnector01 | ● | ● | ● | ● | ● | ● | ● | ● | | ● | | | ● | ● | ● |
| ComparatorActualValueSSIConnector03 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | | | ● | |

### 14.6 Event functions

The module provides configurable event functions. An event function can be connected to physical I/O and the values derived from them (e.g. counters) or be purely used for internal processing.

Every event function has event inputs and outputs. Event functions can also have only inputs or only outputs. Each event output has a unique event ID. It is possible to configure when an event is generated on an event output. The effect of an event is determined by the respective event function.

Event functions can also be linked to one another. The link takes place using the event input. Every event input has a 16-bit register to which the event number of the linked event output is written.

## Information:

**The module functions that can be configured in the Automation Studio I/O configuration are primarily based on these event functions and their links. Changes in the Automation Studio I/O configuration have multiple effects on event functions and their links.**

### 14.6.1 List of event IDs

Various hardware and software functions send event IDs or require event IDs in order to start. The following table shows all of the IDs available to configure the module.

| Event ID | Description | |
|---|---|---|
| **Direct event inputs** | | |
| 512 | Comparator condition 1 | FALSE |
| 513 | | TRUE |
| 544 | Comparator condition 2 | FALSE |
| 545 | | TRUE |
| 576 | Comparator condition 3 | FALSE |
| 577 | | TRUE |
| 608 | Comparator condition 4 | FALSE |
| 609 | | TRUE |
| **Counter comparator functions** | | |
| 2112 | Counter function 1 | Event function 1; FALSE |
| 2113 | | Event function 1; TRUE |
| 2144 | | Event function 2; FALSE |
| 2145 | | Event function 2; TRUE |
| 2368 | Counter function 2 | Event function 1; FALSE |
| 2369 | | Event function 1; TRUE |
| 2400 | | Event function 2; FALSE |
| 2401 | | Event function 2; TRUE |
| 2624 | Counter function 3 | Event function 1; FALSE |
| 2625 | | Event function 1; TRUE |
| 2656 | | Event function 2; FALSE |
| 2657 | | Event function 2; TRUE |
| 2880 | Counter function 4 | Event function 1; FALSE |
| 2881 | | Event function 1; TRUE |
| 2912 | | Event function 2; FALSE |
| 2913 | | Event function 2; TRUE |
| **Edge events** | | |
| 4096 | Falling edge on I/O channel | Channel 1 |
| ... | | ... |
| 4111 | | Channel 16 |
| 4112 | Rising edge on I/O channel | Channel 1 |
| ... | | ... |
| 4127 | | Channel 16 |
| 4128 | Rising or falling edge on I/O channel | Channel 1 |
| ... | | ... |
| 4143 | | Channel 16 |
| **SSI counter events** | | |
| 7168 | SSI 1 | SSI valid |
| 7169 | | SSI ready |
| 7424 | SSI 2 | SSI valid |
| 7425 | | SSI ready |
| **SSI comparator events** | | |
| 7232 | SSI 1 comparator condition | FALSE |
| 7233 | | TRUE |
| 7488 | SSI 2 comparator condition | FALSE |
| 7489 | | TRUE |
| **Timer events** | | |
| 208 | Timer1 | 50 µs |
| 209 | Timer2 | 100 µs |
| 210 | Timer3 | 200 µs |
| 211 | Timer4 | 400 µs |
| 212 | Timer5 | 800 µs |
| 213 | Timer6 | 1600 µs |
| 214 | Timer7 | 3200 µs |
| 215 | Timer8 | 3200 µs (time offset to timer 7) |
| **Network functions** | | |
| 224 | SOAISOP (**s**ynchronous **o**ut **a**synchronous **i**n **s**tart of **p**rotocol) | |
| 225 | AOSISOP (**a**synchronous **o**ut **s**ynchronous **i**n **st**art of **p**rotocol) | |
| 226 | SOAIEOP (**s**ynchronous **o**ut **a**synchronous **i**n **e**nd of **p**rotocol) | |
| 227 | AOSIEOP (**a**synchronous **o**ut synchronous **i**n **e**nd of **p**rotocol) | |
| **Idle event** | | |
| 192 | No-load operation | |

**Timer**

There are 8 timer events that the module can generate.

## Information:

**The timers have the highest event priority. All other system functions are interrupted when a timer event occurs, and jitter for the amount of time it takes to process the event.**

**Idle event**

Idle time is the time that remains after the system has processed all higher priority events and operations. The module performs the following functions during idle time:

- Handling of the asynchronous protocol
- Mechanism for (re-)linking events
- Operation of LEDs
- Execution of event event functions linked to the idle function

### 14.6.2 Edge events

For each physical channel there are 3 event functions

- Falling edge
- Rising edge
- Falling and rising edge

The respective event is triggered when an edge is detected on the hardware input and the "CfO_EdgeDetectRising" on page 24 and/or "CfO_EdgeDetectFalling" on page 23 register has been configured for the respective channel.

Edges are detected by the hardware and processed for each interrupt. The interrupt handler uses an event distributor, which requires a specific amount of time for each edge to operate the hardware and execute linked event functions. To reduce this time, edge detection can be enabled/disabled individually for each channel. To optimize system load and I/O jitter, it is important to only enable edge detection where it is actually needed.

## Information:

**Edge detection can also be used for channels that are configured as outputs.**

#### 14.6.2.1 Event frequency limitation

To stabilize the system, there is a mechanism that limits the number of events created through edge recognition. At least one idle event must occur between two edge events for the same edge.

The "CfO_FallingDisProtection" on page 24 and "CfO_RisingDisProtection" on page 24 registers can be used to disable this limitation for each edge, and then an event will be generated for every edge. However, this can cause a system overload, i.e. I/O operation can fail for up to 100 ms before the module changes to the reset state.

#### 14.6.2.2 Generate event on falling edge

Name:
CfO_EdgeDetectFalling

This register defines whether an event is generated on a falling edge.

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | Channel 1 | 0 | No event generated on falling edge. |
| | | 1 | Events 4096 and 4128 are generated on falling edge. |
| ... | | ... | |
| 15 | Channel 16 | 0 | No event generated on falling edge. |
| | | 1 | Events 4111 and 4143 are generated on falling edge. |

### 14.6.2.3 Generate event on rising edge

Name:
CfO_EdgeDetectRising

This register defines whether an event is generated on a rising edge.

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | Channel 1 | 0 | No event generated on rising edge. |
| | | 1 | Events 4112 and 4128 are generated on rising edge. |
| ... | | ... | |
| 15 | Channel 16 | 0 | No event generated on rising edge. |
| | | 1 | Events 4127 and 4143 are generated on rising edge. |

### 14.6.2.4 Enable limit for falling edges

Name:
CfO_FallingDisProtection

This register can be used to enable/disable the event frequency limit for falling edges on the respective channel.

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | Channel 1 | 0 | Event frequency limit enabled. |
| | | 1 | Event frequency limit disabled. |
| ... | | ... | |
| 15 | Channel 16 | 0 | Event frequency limit enabled. |
| | | 1 | Event frequency limit disabled. |

### 14.6.2.5 Enable limit for rising edges

Name:
CfO_RisingDisProtection

This register can be used to enable/disable the event frequency limit for rising edges on the respective channel.

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | Channel 1 | 0 | Event frequency limit enabled. |
| | | 1 | Event frequency limit disabled. |
| ... | | ... | |
| 15 | Channel 16 | 0 | Event frequency limit enabled. |
| | | 1 | Event frequency limit disabled. |

## 14.6.3 Direct input functions

The module has 2 "direct input functions"

These event functions are based on comparator functionality. If the event configured in the "CfO_DIREKTIOeven-t0IDwr" on page 25 register occurs, the event function compares the status of all Direct I/O channels enabled in the "CfO_EvCompMask" on page 25 register to a status defined in the "CfO_DIREKTIOeventcompState" on page 25 register. The event that is generated depends on the results of this comparison.

- If the respective bits are the same, this is event number 513 or 545.
- If the respective bits are not the same, this is event number 512 or 544.

### 14.6.3.1 Configure event ID for input function

Name:
CfO_DIREKTIOevent0IDwr to CfO_DIREKTIOevent1IDwr

This register holds the event ID generated by the direct input function. For a list of all possible event IDs, see "List of event IDs" on page 22

| Data type | Value | Information[1)] |
|---|---|---|
| UINT | 192 to 7489 | ID of the event function.<br>Bus controller default setting: 4106 |

1) The bus controller default value applies only to the register number specified in function model 254.

### 14.6.3.2 Configure the mode of the input function

Name:
CfO_DIREKTIOevent0mode to CfO_DIREKTIOevent1mode

The mode in which the direct input function operates can be set in this register.

Comparator functions can be operated in 4 different modes. For a description, see "Comparator modes" on page 34.

| Data typ | Value | Bus controller default setting[1)] |
|---|---|---|
| USINT | See bit structure. | 3 |

1) The bus controller default value applies only to the register number specified in function model 254.

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 - 1 | Comparator mode | 0 | Off |
| | | 1 | Individual |
| | | 2 | State change |
| | | 3 | Continuous |
| 2 - 7 | Reserved | - | |

### 14.6.3.3 Comparator status for comparator mask

Name:
CfO_DIREKTIOevent0compState to CfO_DIREKTIOevent1compState

This register contains the status bits that are compared with the bits specified in the "CfO_Ev0CompMask" on page 25 register, which contain the I/O input status, when an event is received.

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | Comparator status of channel 1 | 0 or 1 | |
| ... | | ... | |
| 14 | Comparator status of channel 15 | 0 or 1 | |

### 14.6.3.4 Configure the comparator mask for the input function

Name:
CfO_Ev0CompMask to CfO_Ev1CompMask

If a bit is set, then the input status of the respective channel is compared with that bit in the "CfO_DIREKTIOevent-compState" on page 25 register.

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | Channel 1 | 0 | Do not compare bit |
| | | 1 | Compare bit in register |
| ... | | ... | |
| 14 | Channel 15 | 0 | Do not compare bit |
| | | 1 | Compare bit in register |

### 14.6.4 Direct output functions

The module has 4 of these event functions

The effect of executing this event function is similar to writing to the "DigitalOutput02 to 08" on page 19 registers. When this event function is triggered, however, the changed output states are passed on to the hardware immediately, regardless of the X2X cycle.

When this event function is used, the masks of the respective outputs (see "CfO_OutClearMask" on page 18 and "CfO_OutSetMask" on page 18 registers) must be set to 1. Otherwise the output status would constantly be overwritten by the values in the "DigitalOutput02 to 08" on page 19 registers.

#### 14.6.4.1 Configure event ID for output function

Name:
CfO_DIREKTIOoutevent0IDwr to CfO_DIREKTIOoutevent3IDwr

These registers hold the event IDs that trigger the direct output function. For a list of all possible event IDs, see "List of event IDs" on page 22

| Data type | Value | Information |
|---|---|---|
| INT | 192 to 7489 | ID of event function |

#### 14.6.4.2 Configure channels for resetting

Name:
CfO_DIREKTIOoutclearmask0 to CfO_DIREKTIOoutclearmask3

Writing "1" to the bit position that corresponds to a channel resets the output if the output event function is being executed. This corresponds to writing "0" in register "DigitalOutput" on page 19.

The bit that corresponds to channels that should be reset should be set to "1" in the "CfO_OutClearMask" on page 18 register.

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | Channel 1 | 0 | Reset channel 1 |
| | | 1 | Do not reset channel 1 |
| ... | | ... | |
| 14 | Channel 15 | 0 | Reset channel 15 |
| | | 1 | Do not reset channel 15 |
| 15 | Reserved | - | |

#### 14.6.4.3 Configure channels for setting

Name:
CfO_DIREKTIOoutsetmask0 to CfO_DIREKTIOoutsetmask3

Writing "1" to the bit position that corresponds to a channel sets the output if the output event function is being executed. This corresponds to writing "1" in register "DigitalOutput" on page 19.

The bit that corresponds to channels that should be reset should be set to "1" in the "CfO_OutSetMask" on page 18 register.

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | Channel 1 | 0 | Set channel 1 |
| | | 1 | Do not set channel 1 |
| ... | | ... | |
| 14 | Channel 15 | 0 | Set channel 15 |
| | | 1 | Do not set channel 15 |
| 15 | Reserved | - | |

## 14.7 Counters and encoders

The module has 8 internal counter functions, each with 2 event counter registers. Each of these 8 counters is permanently assigned to 2 physical inputs. This assignment cannot be changed.

The counter registers perform different functions based on how the event functions are connected. The counter registers can be configured in the following ways:

- AB counters
- Up/down counters
- Event counter

Different names are used for them in Automation Studio and in the register description to improve clarity.

| Channel | Counter function | Counter register | Name in Automation Studio |
|---|---|---|---|
| 1 | 1 | 1 | ABConnector01<br>CounterConnector01<br>EventCounter01 |
| 2 | | 2 | EventCounter02 |
| 3 | 2 | 1 | ABConnector02<br>CounterConnector02<br>EventCounter03 |
| 4 | | 2 | EventCounter04 |
| 5 | 3 | 1 | ABConnector03<br>CounterConnector03<br>EventCounter05 |
| 6 | | 2 | EventCounter06 |
| 7 | 4 | 1 | ABConnector04<br>CounterConnector04<br>EventCounter07 |
| 8 | | 2 | EventCounter08 |
| 9 | 5 | 1 | EventCounter09 |
| 10 | | 2 | EventCounter10 |
| 11 | 6 | 1 | EventCounter11 |
| 12 | | 2 | Not used |
| 13 | 7 | 1 | EventCounter13 |
| 14 | | 2 | EventCounter14 |
| 15 | 8 | 1 | EventCounter15 |
| 16 | | 2 | Not used |

### 14.7.1 Counter value calculation

There are 3 steps for calculating the state of any counter function

1. The counter value is based on the 2 absolute value counters "abs1" and "abs2". These are only used internally in the module and cannot be read. Depending on the mode, these registers show the respective physical input signals.

| | Mode | | |
|---|---|---|---|
| | Edge counters | AB counters | Up/Down counters |
| abs1 | Edges of counter channel 1 | Increments in positive direction | Counter channel 2 = 0:<br>Edges of counter channel 1<br>in up direction |
| abs2 | Edges of counter channel 2 | Increments in negative direction | Counter channel 2 = 1<br>Edges of counter channel 1<br>in down direction |

2. 2 more counters are formed from absolute value registers "abs1" and "abs2": "counter1" and "counter2". They are only used internally in the module and cannot be read. The following values are used for the calculation:

- Absolute value registers "abs1" and "abs2"
- SW_reference_counter 1 and 2: This reference value can be predefined by register "CfO_CounterPreset-Value" on page 32 to allow referencing <> 0.
- HW_reference_counter 1 and 2: In register "CfO_CounterEventMode" on page 35, you can configure whether latched values should be copied to these registers when counter events occur.

counter1 = abs1 + SW_reference_counter1 - HW_reference_counter1
counter2 = abs2 + SW_reference_counter2 - HW_reference_counter2

3. The counter registers contain the sum of the two internal counters "counter 1" and "counter 2". The "CfO_Coun-terConfigReg" on page 31 register allows you to define a sign for each "counter" register and define whether or not it should be used.

Counter register = counter1 + counter2

### 14.7.2 Sample configurations

All of the settings available in Automation Studio for ABR encoders, AB counters, up/down counters and event counters are based on the 2 counter functions.
The following configuration examples show the values with which Automation Studio initializes the module registers in order to implement these functions.

#### 14.7.2.1 I/O configuration - ABR encoder

The following table shows how the module's various event functions can be linked in order to configure an ABR encoder.

| Register | Value | Comment |
|---|---|---|
| **For the function** | | |
| CfO_Counter5PresetValue1<br>CfO_Counter7PresetValue1 | (any) | Desired offset value for referencing |
| CfO_Counter5event0IDwr<br>CfO_Counter7event0IDwr | 0x0201 | Link between the first counter event and the "direct input" comparator condition TRUE |
| CfO_Counter5config<br>CfO_Counter7config | 0x01 | Mode = AB counter |
| CfO_Counter5configReg0<br>CfO_Counter7configReg0 | 0x0D | Configures the calculation of the internal registers "counter1" and "counter2" (see "Counter value calculation" on page 27 and "Examples of calculation configurations" on page 31) |
| CfO_DIREKTIOevent0IDwr<br>CfO_DIREKTIOevent1IDwr | 0x1002 or 0x1012 | Selection of the desired input edge as trigger for the ABR encoder function |
| CfO_Counter5event0config<br>CfO_Counter7event0config | 0x0000 | Configuration of the first counter event (for referencing) |
| CfO_DIREKTIOevent0mode<br>CfO_DIREKTIOevent1mode | 0x03 | Mode of the "direct input function" - Continuous |
| CfO_DIREKTIOevent0compState<br>CfO_DIREKTIOevent1compState | 0x00 or 0x08 | Comparator status for the "direct input function" |
| CfO_Ev0CompMask<br>CfO_Ev1CompMask | 0x08 | Comparator mask for the "direct input function" |
| **For the latch** | | |
| CfO_Counter5event0config<br>CfO_Counter7event1config | 0x000D | Configuration of the calculation of the value used for the latch |
| CfO_Counter5event0mode<br>CfO_Counter7event1mode | 0x03 | Mode of the first counter event function - Continuous |
| CfO_Counter5event0IDwr<br>CfO_Counter7event1IDwr | (any) | Number of the event that should trigger the latch |
| **For the comparator** | | |
| CfO_Counter5event1IDwr<br>CfO_Counter7event1IDwr | 0x00D0 | Event number of Timers1 (50 µs)<br>**Information:**<br>**The latch and comparator must not have the same event number!** |
| CfO_Counter5event1config<br>CfO_Counter7event1config | 0x900D or 0xA00D | Configuration of the comparator for the second counter event |
| CfO_DIREKTIOoutevent0IDwr<br>CfO_DIREKTIOoutevent2IDwr | 0x0861<br>0x0A61 | TRUE event output of the second counter to trigger the direct output function (set outputs). |
| CfO_DIREKTIOoutsetmask0<br>CfO_DIREKTIOoutsetmask2 | 0x08, 0x20, 0x80<br>0x02, 0x08, 0x80 | Outputs that should be set when comparator condition = TRUE |
| CfO_DIREKTIOoutevent1IDwr<br>CfO_DIREKTIOoutevent3IDwr | 0x0860<br>0x0A60 | FALSE event output of the second counter to trigger the direct output function (reset outputs). |
| CfO_DIREKTIOoutclearmask1<br>CfO_DIREKTIOoutclearmask3 | 0x08, 0x20, 0x80<br>0x02, 0x08, 0x80 | Outputs that should be reset when comparator condition = FALSE |

### 14.7.2.2 I/O configuration - AB counter

The following table shows how the module's various event functions can be linked in order to configure an AB counter.

[x] stands for the respective counter function, from 1 to 4

| Register | Value | Comment |
|---|---|---|
| **For the function** | | |
| CfO_Counter[x]config | 0x01 | Mode = Up/down counter |
| CfO_Counter[x]configReg0 | 0x0D | Configures the calculation of the internal registers "counter1" and "counter2" (see "Counter value calculation" on page 27 and "Examples of calculation configurations" on page 31) |
| **For the latch** | | |
| CfO_Counter[x]event0config | 0x000D | Configuration of the calculation of the first value used for the latch |
| CfO_Counter[x]event0mode | 0x03 | Mode of the first counter event function - Continuous |
| CfO_Counter[x]event0IDwr | (any) | Number of the event that should trigger latch 1 ("Latch 01 - Channel" in the Automation Studio I/O configuration). |
| CfO_Counter[x]event1config | 0x0D | Configuration of the calculation of the second value used for the latch |
| CfO_Counter[x]event1mode | 0x03 | Mode of the second counter event function - Continuous |
| CfO_Counter[x]event1IDwr | (any) | Number of the event that should trigger latch 2 |
| **For the comparator** | | |
| CfO_Counter1event1IDwr<br>CfO_Counter3event1IDwr | 0x00D0 | Event number of Timer 1 (50 µs)<br>**Information:**<br>**The latch and comparator must not have the same event number!** |
| CfO_Counter1event1config<br>CfO_Counter3event1config | 0x900D or 0xA00D | Configuration of the comparator for the second counter event |
| CfO_Counter1event1mode<br>CfO_Counter3event1mode | 0x03 | Mode of the second counter event function - Continuous |
| CfO_DIREKTIOoutevent0IDwr<br>CfO_DIREKTIOoutevent2IDwr | 0x0861<br>0x0A61 | TRUE event output of the second counter to trigger the direct output function (set outputs). |
| CfO_DIREKTIOoutsetmask0<br>CfO_DIREKTIOoutsetmask2 | 0x08, 0x20, 0x80<br>0x02, 0x08, 0x80 | Outputs that should be set when comparator condition = TRUE |
| CfO_DIREKTIOoutevent1IDwr<br>CfO_DIREKTIOoutevent3IDwr | 0x0860<br>0x0A60 | FALSE event output of the second counter to trigger the direct output function (reset outputs). |
| CfO_DIREKTIOoutclearmask1<br>CfO_DIREKTIOoutclearmask3 | 0x08, 0x20, 0x80<br>0x02, 0x08, 0x80 | Outputs that should be reset when comparator condition = FALSE |

### 14.7.2.3 I/O configuration - Up/down counter

The following table shows how the module's various event functions can be linked in order to configure an up/down counter.

[x] stands for the respective counter function, from 1 to 4

| Register | Value | Comment |
|---|---|---|
| **For the function** | | |
| CfO_Counter[x]config | 0x03 | Counter mode = Up/down counter |
| CfO_Counter[x]configReg0 | 0x0D, 0x07 | Configures the calculation of the internal registers "counter1" and "counter2" (see "Counter value calculation" on page 27 and "Examples of calculation configurations" on page 31) |
| **For the latch** | | |
| CfO_Counter[x]event0config | 0x0D, 0x07 | Configuration of the calculation of the first value used for the latch |
| CfO_Counter[x]event0mode | 0x03 | Mode of the first counter function - Continuous |
| CfO_Counter[x]event0IDwr | (any) | Number of the event that should trigger Latch 1 |
| CfO_Counter[x]event1config | 0x0D, 0x07 | Configuration of the calculation of the second value used for the latch |
| CfO_Counter[x]event1mode | 0x03 | Mode of the second counter function - Continuous |
| CfO_Counter[x]event1IDwr | (any) | Number of the event that should trigger Latch 2 |
| **For the comparator** | | |
| CfO_Counter1event1IDwr<br>CfO_Counter3event1IDwr | 0x00D0 | Event number of Timer 1 (50 µs)<br>**Information:**<br>**The latch and comparator must not have the same event number!** |
| CfO_Counter1event1config<br>CfO_Counter3event1config | 0x900D, 0xA00d or 0x9007, 0xA007 | Configuration of the comparator for the second counter event |
| CfO_Counter1event1mode<br>CfO_Counter3event1Imode | 0x03 | Mode of the second counter event function - Continuous |
| CfO_DIREKTIOoutevent0IDwr<br>CfO_DIREKTIOoutevent2IDwr | 0x0861 | TRUE event output of the second counter to trigger the direct output function (set outputs). |
| CfO_DIREKTIOoutsetmask0<br>CfO_DIREKTIOoutsetmask2 | 0x08, 0x20, 0x80<br>0x02, 0x08, 0x80 | Outputs that should be set when comparator condition = TRUE |
| CfO_DIREKTIOoutevent1IDwr<br>CfO_DIREKTIOoutevent3IDwr | 0x0860<br>0x0A60 | FALSE event output of the second counter to trigger the direct output function (reset outputs). |
| CfO_DIREKTIOoutclearmask1<br>CfO_DIREKTIOoutclearmask3 | 0x08, 0x20, 0x80<br>0x02, 0x08, 0x80 | Outputs that should be reset when comparator condition = FALSE |

#### 14.7.2.4 I/O configuration - Event counter

The following table shows how the module's various event functions can be linked in order to configure an event counter.

[x] stands for the respective counter function, from 1 to 8

| Register | Value | Comment |
|---|---|---|
| **For event counters on uneven channel numbers (counter register 1)** | | |
| CfO_Counter[x]configReg0 | 0x01 or 0x03 | Configures the calculation of the internal registers "counter1" and "counter2" (see "Counter value calculation" on page 27 and "Examples of calculation configurations" on page 31) |
| CfO_Counter[x]event0mode | 0x43 | Mode of the first counter event function and referencing configuration |
| CfO_Counter[x]event0IDwr | (any) | Number of the event that should trigger referencing |
| **For event counters on even channels (counter register 2)** | | |
| CfO_Counter[x]configReg1 | 0x04 or 0x08 | Configures the calculation of the internal registers "counter1" and "counter2" (see "Counter value calculation" on page 27 and "Examples of calculation configurations" on page 31) |
| CfO_Counter[x]event1mode | 0x83 | Mode of the second counter event function and referencing configuration |
| CfO_Counter[x]event1IDwr | (any) | Number of the event that should trigger referencing |

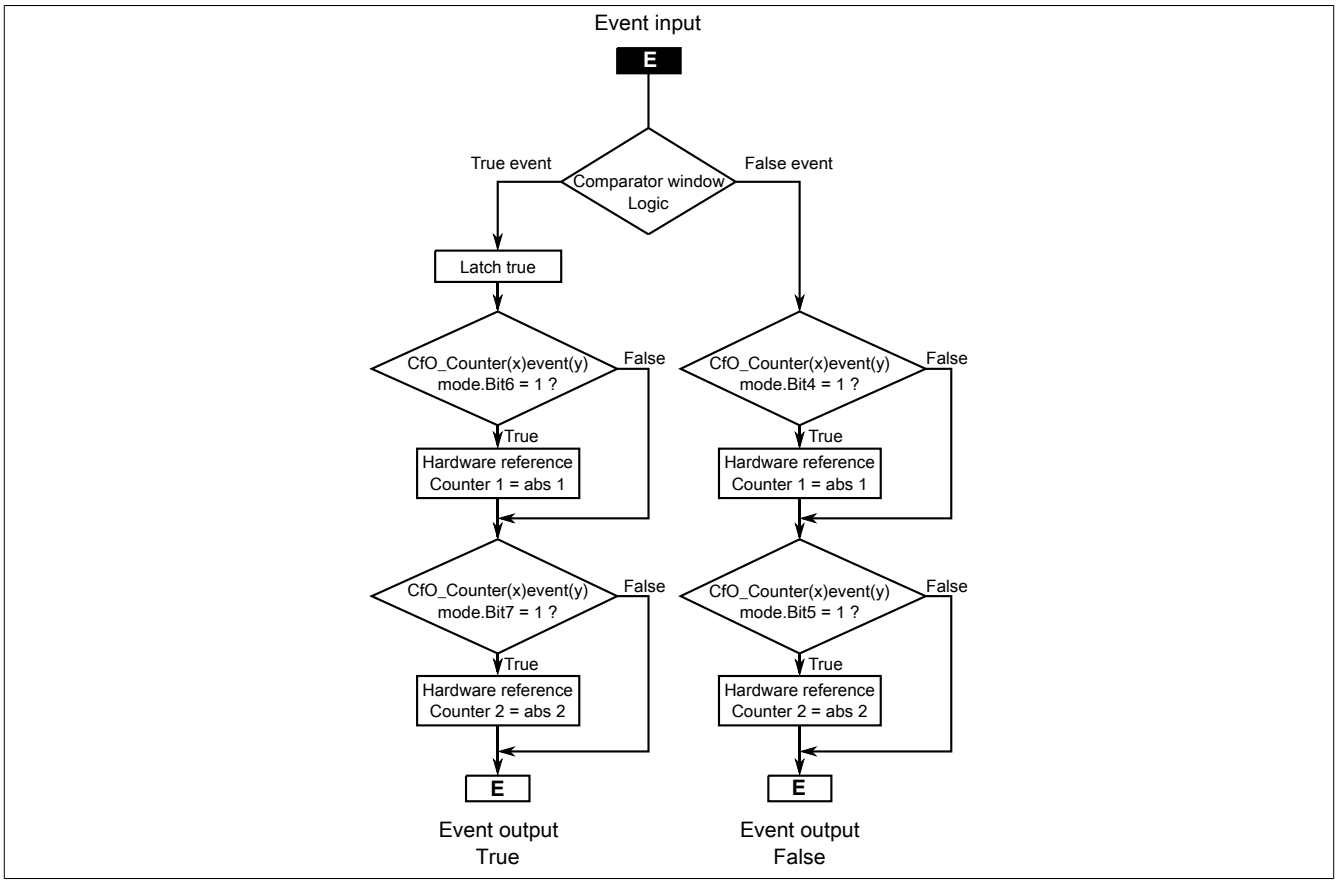
### 14.7.3 General event functions

Each of the 8 counter functions has 2 counter event functions. These consist of:

- Event ID that triggers the counter event function
- A window comparator
- Latch register for saving the counter value

When the counter event function is complete, a combined event ID in the range 2112 to 2913 (see "List of event IDs" on page 22) is sent.

Each counter event function also has the option to copy the current counter value to the "reference counter" when an event occurs (see "Counter value calculation" on page 27).



### 14.7.3.1 Configure counter mode

Name:
CfO_Counter1config to CfO_Counter8config

These registers are used to configure the mode of the counter function. Each counter function can be operated in 3 different modes.

| | Counter function mode | | |
|---|---|---|---|
| | **Edge counters** | **AB counters** | **Up/down counter** |
| Counter channel 1[1)] | Counting pulses, edge counter 1 | A | Metering pulses |
| Counter channel 2[1)] | Counting pulses, edge counter 2 | B | Counting direction (0 = positive, 1 = negative) |
| Counter register 1 | Counter value 1 | Position | Counter value |
| Counter register 2 | Counter value 2 | | |

1) Corresponds to the physical channels of the counter functions. See "Description of channel assignments" on page 11

| Data type | Values | Bus controller default setting[1)] |
|---|---|---|
| USINT | See bit structure. | CfO_Counter**N**config<br>N(1,2,3,5): 1<br>N(4): 0 |

1) The bus controller default value applies only to the register numbers specified in function model 254.

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 - 1 | Counter mode | 00 | Edge counter (bus controller default setting N(4)) |
| | | 01 | AB counter (bus controller default setting (N(1,2,3,5)) |
| | | 11 | Up/Down counters |
| 2 - 7 | Reserved | - | |

### 14.7.3.2 Configure calculation of internal counters

Name:
CfO_Counter1configReg0 to CfO_Counter8configReg0 ("counter1")
CfO_Counter1configReg1 to CfO_Counter8configReg1 ("counter2")

The calculation of the internal "counter1" and "counter2" registers can be configured in these registers. For information on using these internal registers, see "Counter value calculation" on page 27.

| Data type | Values | Bus controller default setting[1)] |
|---|---|---|
| USINT | See bit structure. | CfO_Counter**N**configReg0<br>N(1,2,3,5): 13<br>N(4): 0<br>CfO_Counter**N**configReg1<br>N(1,2,3,5): 0<br>N(0): 4 |

1) The bus controller default value applies only to the register numbers specified in function model 254.

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | counter1 - Use | 0 | 0 is added instead of register "counter1". |
| | | 1 | "counter1" is used for addition. |
| 1 | counter1 - Sign | 0 | The sign of register "counter1" is not changed for addition. |
| | | 1 | The sign of register "counter1" is reversed for addition. |
| 2 | counter2 - Use | 0 | 0 is added instead of register "counter2". |
| | | 1 | "counter2" is used for addition. |
| 3 | counter2 - Sign | 0 | The sign of register "counter2" is not changed for addition. |
| | | 1 | The sign of register "counter2" is reversed for addition. |
| 4 - 7 | Reserved | - | |

## Examples of calculation configurations

| | | |
|---|---|---|
| 0b00000001 | = 0x01 | Only the "counter1 - use" bit is set, entering the contents of the "counter" (edge of counter event channel 1) directly in the counter register. |
| 0b00000011 | = 0x03 | "counter 1 - use" and "counter1 - sign" bits are set. The sign is changed so that the counter register counts in the negative direction. |
| 0b00001101 | = 0x0d | Edges on counter input channel 1 increase the value in the counter register. Edges on counter input channel 2 decrease the value in the counter register. This value is the best setting for modes "AB counter" and "Up/Down counter". |

### 14.7.3.3 Offset value for referencing

Name:
CfO_Counter1PresetValue1 to CfO_Counter8PresetValue1
CfO_Counter1PresetValue1_32Bit to CfO_Counter8PresetValue1_32Bit (SW_reference_counter1)

CfO_Counter1PresetValue2 to CfO_Counter8PresetValue2
CfO_Counter1PresetValue2_32Bit to CfO_Counter8PresetValue2_32Bit (SW_reference_counter2)

These registers can be used to define an offset value for referencing. This value is copied to internal register SW_reference_counter of the respective counter register.

| Data type | Values | Information[1)] |
|---|---|---|
| INT | -32768 to 32767 | Bus controller default setting: 0 |
| DINT | -2,147,483,648 to 2,147,483,647 | |

1) The bus controller default value applies only to the register numbers specified in function model 254.

### 14.7.3.4 Counter registers

Name:
Different names are used for these 16 registers in Automation Studio and in the register description depending on the function.

ABConnector01 to ABConnector04
CounterConnector01 to CounterConnector04
EventCounter01 to EventCounter15
ABRConnector01 and ABRConnector03

These 16 registers show the results of the counter value calculation for the respective register. Depending on the function, this corresponds to either the encoder position or the counter value.

For information on the relationship between physical channels and counter registers, see "Counters and encoders" on page 27 and "Description of channel assignments" on page 11

| Data type | Value | Information |
|---|---|---|
| INT | -32,768 to 32,767 | Encoder position or counter value |
| DINT[1)] | -2,147,483,648 to 2,147,483,647 | Encoder position or counter value |

1) Only in function model 1

### 14.7.3.5 Status of the ABR encoder

Name:
StatusABRConnector01 to StatusABRConnector02

The referencing status of the ABR encoder is indicated in this register.

| Data type | Values |
|---|---|
| USINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 - 1 | Reserved | 0 | |
| 2 | Bit is always 1 after the first reference pulse. | 0 | No reference pulses have occurred since the start of referencing. |
| | | 1 | The first reference pulse has occurred. |
| 3 | State change when referencing is complete | 0 or 1 | State change when referencing is complete |
| 4 | Bit is always 1 after the first reference pulse. | 0 | No reference pulses have occurred since the start of referencing. |
| | | 1 | The first reference pulse has occurred. |
| 5 - 7 | Continuous counter | xxx | Increased with each reference pulse |

### Examples of possible values

| | | |
|---|---|---|
| 0b00000000 | = 0x00 | Referencing OFF or homing procedure already active |
| 0b00111100 | = 0x3C | First reference complete, reference value applied in the "ABREncoder0" on page 32 register |
| 0bxxx11100 | = 0xxB | Bits 5 to 7 are changed with each reference pulse |
| 0bxxx1x100 | = 0xxx | Bits changed continuously with the setting continuous referencing. With every reference pulse, the reference value is applied to the "ABREncoder0" on page 32 register |

### 14.7.3.6 Configure ABR referencing mode

Name:
ReferenceModeABRConnector01 and ReferenceModeABRConnector03

The bits in this register are used to configure the reaction to the configured reference pulse.

| Data type | Values |
|---|---|
| USINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 - 1 | Sets the referencing mode | 00 | Referencing OFF |
| | | 01 | Single-shot referencing |
| | | 10 | Reserved |
| | | 11 | Continuous referencing |
| 2 - 5 | Reserved | - | |
| 6 - 7 | Reserved | 11 | Must always be 11! |

This results in the following values:

| | | |
|---|---|---|
| 0b00000000 | = 0x00 | Referencing OFF |
| 0b11000001 | = 0xC1 | Single-shot referencing →When starting over after the referencing process is complete, the value 0x00 must be written to start again. Wait until the "StatusABR" on page 32 register also takes on the value 0x00, then the value 0xC1 can be written again. |
| 0b11000011 | = 0xC3 | Continuous referencing → Referencing takes place automatically with every reference pulse |

### 14.7.4 Comparator functions

The ABR and AB counters and the up/down counter have a comparator function. It always works the same and is described here globally for all three.

The comparators are implemented in software form. They do not work actively but rather passively, i.e. the comparison is only carried out when an event is received. The event received is forwarded along the TRUE or FALSE branch depending on the status of the comparator condition. An event function like this generally also offers a latch for the TRUE and FALSE branch to save the value used for the comparator at the time of the event.

#### 14.7.4.1 Comparator modes

Comparator functions can be operated in 4 different modes.

- **Off**
  Events are ignored.
- **Individual**
  The event function is executed once and then disables itself automatically. To re-enable it, the "event function mode" must be changed, preferably to "off" and then to the desired mode. This setting allows a hardware latch to be simulated.
- **State change**
  The event function only responds when the comparator status has changed, i.e. from false to true (or vice versa). Only the first event for each status is processed, e.g. the first "true" of a sequence of events with the comparator condition "true". After the event function is enabled, the first incoming event is used to determine the starting status and therefore not forwarded. This setting allows a hardware comparator to be simulated.
- **Continuous**
  Each incoming event is forwarded to the true or false branch depending on the comparator condition. This setting allows event filters to be created.

#### 14.7.4.2 Configure event ID for comparator

Name:
CfO_Counter1event0IDwr to CfO_Counter8event0IDwr (event function 1)
CfO_Counter1event1IDwr to CfO_Counter8event1IDwr (event function 2)

This register holds the event ID that should trigger the counter event function. For a list of all possible event IDs, see "List of event IDs" on page 22

| Data type | Value | Information |
|---|---|---|
| UINT | 192 to 7489 | ID of the counter event function.<br>Bus controller default setting: 513 |

1) The bus controller default value applies only to the register number specified in function model 254.

### 14.7.4.3 Configure calculation of comparator

Name:
CfO_Counter1event0config to CfO_Counter8event0config (event function 1)
CfO_Counter1event1config to CfO_Counter8event1config (event function 2)

These registers are used to configure the counter event function for the respective counter function.

Bits 0 to 3 configure the calculation of the comparison or to latch the value. This calculation is similar to the calculation of the counter register (see "Counter value calculation" on page 27)

Bits 8 to 13 can be used to limit the number of bits used for the comparison. A mask is calculated as $2^n$ - 1 and linked with an "AND" operation. This makes it possible to generate a comparator pulse every $2^n$ increments.

| Data type | Values | Bus controller default setting[1)] |
|---|---|---|
| UINT | See bit structure. | 0 |

1) The bus controller default value applies only to the register number specified in function model 254.

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | counter 1 - use | 0 | 0 is added instead of "counter 1" |
| | | 1 | "counter 1" is used for addition |
| 1 | counter 1 - sign | 0 | The sign of the "counter 1" register is not changed for addition |
| | | 1 | The sign of the "counter 1" register is reversed for addition |
| 2 | counter 2 - use | 0 | 0 is added instead of register "counter2". |
| | | 1 | "counter 2" is used for addition |
| 3 | counter 1 - sign | 0 | The sign of the "counter 2" register is not changed for addition |
| | | 1 | The sign of the "counter 2" register is reversed for addition |
| 4 - 7 | Reserved | - | |
| 8 - 13 | Number of bits for comparator mask | x | The mask value is calculated as $2^n$-1, where n is value set in these bits. |
| 14 | Reserved | - | |
| 15 | Margin comparator mode | 0 | MarginComparator >= (Current position - OriginComparator) |
| | | 1 | MarginComparator > (Current position - OriginComparator) |

### 14.7.4.4 Configure mode and latching of comparator function

Name:
CfO_Counter1event0mode to CfO_Counter8event0mode (event function 1)
CfO_Counter1event1mode to CfO_Counter8event1mode (event function 2)

In these registers you can set the mode for the comparator function and optional copying of the latched registers.

Comparator functions can be operated in 4 different modes. For a description, see "Comparator modes" on page 34.

Bits 4 to 7 can be used to define hardware referencing actions.

Based on these bits, the values of the internal absolute value counters "abs1" and "abs2" can be copied to the respective "HW_reference_counter" register at every counter event (see "Counter value calculation" on page 27). This function can be used to reference the counter values directly in the hardware.

| Data type | Values |
|---|---|
| USINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 - 1 | Comparator mode | 0 | Off |
| | | 1 | Individual |
| | | 2 | State change |
| | | 3 | Continuous |
| 2 - 3 | Reserved | - | |
| 4 | Copy abs1 counter value | 0 | No action |
| | | 1 | When event is FALSE → hardware reference counter 1 = abs1 |
| 5 | Copy abs2 counter value | 0 | No action |
| | | 1 | When event is FALSE → hardware reference counter 2 = abs2 |
| 6 | Copy abs1 counter value | 0 | No action |
| | | 1 | When event is TRUE → hardware reference counter 1 = abs1 |
| 7 | Copy abs2 counter value | 0 | No action |
| | | 1 | When event is TRUE → hardware reference counter 2 = abs2 |

### 14.7.4.5 Width of the comparator

Name:
MarginComparatorABConnector01 to MarginComparatorABConnector04
MarginComparatorABRConnector01 and MarginComparatorABRConnector03
MarginComparatorCounterConnector01 to MarginComparatorCounterConnector04

This register is available for the comparator function of the ABR encoder, AB counter and up/down counter.

It defines the width of the comparator window in the positive direction.

| Data type | Value | Information |
|---|---|---|
| INT | -32768 to 32767 | Width of comparator window, 16-bit |
| DINT | -2,147,483,648 to 2,147,483,647 | Width of comparator window, 32-bit |

### 14.7.4.6 Comparator origin

Name:
OriginComparatorABConnector01 to OriginComparatorABConnector04
OriginComparatorABRConnector01 and OriginComparatorABRConnector03
OriginComparatorCounterConnector01 to OriginComparatorCounterConnector04

This register is available for the comparator function of the ABR encoder, AB counter and up/down counter.

It defines the position value at which the respective configured comparator output channel is set.

| Data type | Value | Information |
|---|---|---|
| INT | -32,768 to 32,767 | Comparator window origin, 16-bit |
| DINT | -2,147,483,648 to 2,147,483,647 | Comparator window origin, 32-bit |

### 14.7.4.7 Read latch position or counter value

Name:
Latch01ABConnector01 to Latch01ABConnector04 (event function 1)
Latch02ABConnector01 to Latch02ABConnector04 (event function 2)
Latch01ABRConnector01 and Latch01ABRConnector03
Latch01CounterConnector01 and Latch01CounterConnector04 (event function 1)
Latch02CounterConnector01 and Latch02CounterConnector04 (event function 2)

These registers are available for the comparator function of the ABR encoder, AB counter and up/down counter.

If the comparator returns "TRUE", then the current counter value is latched and copied to these registers. The calculation of the comparator value used for the latch can be configured in the "CfO_Counter[x]event[y]config" on page 35 register.

| Data type | Value | Information |
|---|---|---|
| INT | -32,768 to 32,767 | Latched encoder position or counter value |
| DINT[1)] | -2,147,483,648 to 2,147,483,647 | Latched encoder position or counter value |

1) Only in function model 1

## 14.8 SSI encoder interface

The module has 2 SSI encoders available, supported directly in the hardware. 2 5 V output channels are set for each SSI encoder and cannot be changed. (See also "Description of channel assignments" on page 11)

When using the SSI encoder, the corresponding clock channel can be configured in the "CfO_CFGchannel" on page 17 register as "Channel-specific" and "Push/Pull".

| Encoder | Data channel | Clock channel |
|---|---|---|
| SSI1 | 9 | 11 |
| SSI2 | 13 | 15 |

### 14.8.1 SSI event functions

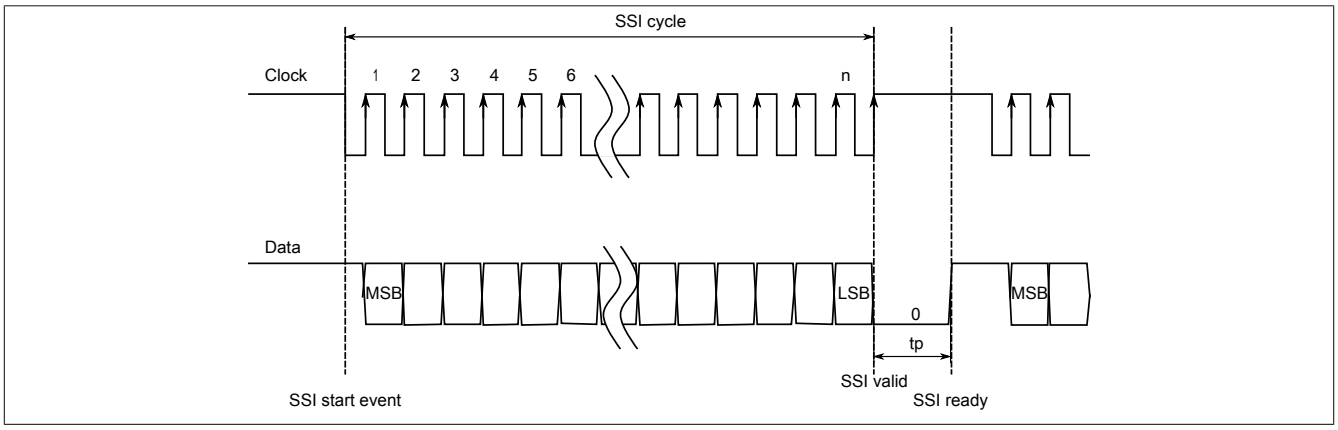
Each of the 2 SSI encoders consists of an event function and an event input. The SSI cycle is started when an event is received on this input.

> **Information:**
>
> **The SSI event function is not linked to an event by default, i.e. SSI functions are disabled.**

2 events are sent from the SSI encoder interface..

- An "SSI valid" event is triggered immediately after the end of the SSI cycle if a new counter value is available.
- The "SSI ready" event then shows when the monoflop time has expired (tp in SSI encoder timing diagram). This is the earliest that the next SSI cycle can be started.

**SSI encoder - Timing diagram**




#### 14.8.1.1 Configure event ID for SSI

Name:
CfO_SSI1eventIDwr to CfO_SSI2eventIDwr

This register holds the event ID that should start the SSI cycle. For a list of all possible event IDs, see "List of event IDs" on page 22

Normally this register is set to network event 225 "AOSISOP"- This ensures that the new encoder position is available at the next "I/O → Synchronous Frame" transfer. Check the SSI transfer time and the X2X cycle time, because the SSI cycle must be completed within this time.

| Data type | Value | Information[1)] |
|---|---|---|
| UINT | 192 to 7489 | ID of the event function.<br>Bus controller default setting: 225 |

1) The bus controller default value applies only to the register number specified in function model 254.

#### 14.8.1.2 Configure SSI

Name:
CfO_SSI1cfg to CfO_SSI2cfg

This configuration register sets the encoding, clock rate and number of bits.

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 - 5 | SSI value valid bits | x | |
| 6 - 7 | Clock rate | 00 | 1 MHz |
| | | 01 | 500 kHz |
| | | 10 | 250 kHz |
| | | 11 | 125 kHz |
| 8 - 13 | SSI number of bits | x | Number of bits including leading zeros |
| 14 | Reserved | 0 | |
| 15 | Keying | 0 | Binary encoding |
| | | 1 | Gray encoding |

#### 14.8.1.3 SSI advanced configuration

Name:
ConfigAdvanced01 to ConfigAdvanced02

This configuration register is used to set the coding, the clock rate, the number of bits and the monostable multivibrator settings.

It only differs from "CfO_SSI1cfg" on page 38 by data length and additional monostable multivibrator testing.

| Data type | Value | Bus controller default setting[1)] |
|---|---|---|
| UDINT | See bit structure. | 0x10000 |

1) The bus controller default value applies only to the register number specified in function model 254.

Bit structure:

| Bit | Name | Value | Information |
|---|---|---|---|
| 0 - 5 | SSI value valid bits | x | Bus controller default setting: 0 |
| 6 - 7 | Clock rate | 00 | 1 MHz (bus controller default setting) |
| | | 01 | 500 kHz |
| | | 10 | 250 kHz |
| | | 11 | 125 kHz |
| 8 - 13 | SSI number of bits | x | Number of bits including leading zeros.<br>Bus controller default setting: 0 |
| 14 | Reserved | 0 | |
| 15 | Encoding | 0 | Binary encoding (bus controller default setting) |
| | | 1 | Gray encoding |
| 16 - 17 | Monostable multivibrator check | 00 | Check OFF, no additional clock bit |
| | | 01 | Check set to high level (bus controller default setting) |
| | | 10 | Check set to low level |
| | | 11 | Level is clocked but ignored |
| 18 - 31 | Reserved | 0 | |

#### 14.8.1.4 Enable SSI event function

Name:
CfO_SSI1control to CfO_SSI2control

The 2 SSI encoder events can be enabled/disabled using this register.

| Data type | Values |
|---|---|
| USINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | Event: "SSI valid" | 0 | Not transmitted |
| | | 1 | Sent |
| 1 | Event: "SSI ready" | 0 | Not sent |
| | | 1 | Sent |
| 2 - 7 | Reserved | - | |

#### 14.8.1.5 Read SSI position

Name:
SSIConnector01 and SSIConnector03

The last transferred SSI position can be read out from this register. The SSI encoder value is displayed as a 32-bit position value. This position value is generated synchronously with the X2X cycle.

| Data type | Value | Information |
|---|---|---|
| UDINT | 0 to 4,294,967,295 | Last SSI position transferred |

### 14.8.2 SSI comparator condition

The module has an assigned comparator function for the SSI function. These consist of:

- Event ID that triggers the comparator function
- The window comparator
- Latch register for saving the SSI position

When the comparator function is complete, event ID 7232 to7489 (see "List of event IDs" on page 22) is sent.

#### 14.8.2.1 Configure event ID for SSI comparator

Name:
CfO_SSI1event0IDwr to CfO_SSI2event0IDwr

This register holds the event ID that should start the SSI comparator function. For a list of all possible event IDs, see "List of event IDs" on page 22

| Data type | Value | Information |
|---|---|---|
| INT | 192 to 7489 | ID of comparator function |

#### 14.8.2.2 Configure the mode of the SSI comparator function

Name:
CfO_SSI1event0mode to CfO_SSI2event0mode

This register can be used to configure the mode of the comparator function.

Comparator functions can be operated in 4 different modes. For a description, see "Comparator modes" on page 34.

| Data type | Values | |
|---|---|---|
| USINT | See the bit structure. | |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 - 1 | Comparator mode | 0 | Off |
| | | 1 | Individual |
| | | 2 | State change |
| | | 3 | Continuous |
| 2 - 7 | Reserved | - | |

### 14.8.2.3 Configure calculation of SSI comparator

Name:
CfO_SSI1event0config and CfO_SSI2event0config

The calculation of the position value used for the comparator can be configured in this register.

The window comparator condition is calculated as follows:

```
counter_window_value = ssi_counter & (2^ssi_data_bits - 1)
diff = counter_window_value – origin_comparator
if ((diff & (2^(comparator_mask)-1)) <= margin_comparator)
condition = True;
else
condition = False;
```

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 - 5 | SSI data bits | x | Number of data bits used for masking |
| 6 - 7 | Reserved | - | |
| 8 - 13 | Comparator mask | x | The mask value is calculated from $2^n$-1, where n is the value configured in SSI data bits. Default: 0 |
| 14 | Comparator mode | 0 | MarginComparator >= SSI position - OriginComparator |
| | | 1 | MarginComparator > SSI position - OriginComparator |

### 14.8.2.4 Origin of the SSI comparator

Name:
OriginComparatorSSIConnector01 and OriginComparatorSSIConnector03

This register contains the origin of the window comparator.

| Data type | Value | Information |
|---|---|---|
| UDINT | 0 to 4,294,967,295 | Origin of the window comparator. |

### 14.8.2.5 Width of the SSI comparator

Name:
MarginComparatorSSIConnector01 and MarginComparatorSSIConnector03

This register provides the width of the window comparator.

| Data type | Value | Information |
|---|---|---|
| UDINT | 0 to 4,294,967,295 | Width of the SSI window comparator |

### 14.8.2.6 Read SSI latch position

Name:
Latch01SSIConnector01 and Latch01SSIConnector03

If the SSI window comparator returns "True", then the current SSI position is latched and saved in this register.

| Data type | Value | Information |
|---|---|---|
| UDINT | 0 to 4,294,967,295 | Latched SSI position |

## 14.9 PWM - Pulse width modulation

The module has 2 PWM functions available, supported directly by the hardware. A 24 V output channel is set for each PWM encoder and cannot be changed. (See also "Description of channel assignments" on page 11)

When using the PWM function, the corresponding channel can be configured in the "CfO_CFGchannel" on page 17 register as "Channel-specific".

| PWM function | Channel |
|---|---|
| PWM1 | 3 |
| PWM2 | 7 |

### 14.9.1 Configure PWM prescaler

Name:
CfO_PWM0prescaler to CfO_PWM1prescaler

The length of the PWM cycle is configured using this register. The base is a 48 MHz clock, which can be changed (divided) using the setting in this register. One PWM cycle consists of 1000 of the resulting clocks after they have been divided. The period duration of the PWM cycle is calculated as follows:

$$PWM\_cycle = 1000 \frac{prescale}{48000000} [s]$$

| Data type | Value | Information[1)] |
|---|---|---|
| UINT | 2 to 65535 | Prescaler for PWM cycle.<br>Bus controller default setting: 480 |

1) The bus controller default value applies only to the register number specified in function model 254.

## Information:

**The period duration of the PWM function must be greater than 500 µs. Period durations that are too short cause the outputs to heat up considerably.**

### 14.9.2 Output PWM values

Name:
PWMOutput03, PWMOutput07

In this register, a configuration is made for the percentage of the PWM cycle (in 1/10% steps) that the PWM output is logical 1, i.e. ON.

| Data type | Value | Information |
|---|---|---|
| UINT | 0 | PWM output always off |
| | 1 to 999 | Turn on time in 1/10% steps |
| | 1000 | PWM output always on |

## 14.10 Time measurement function

The module has a time measurement function for each I/O channel. It can be configured separately for rising and falling edges on each channel.
A starting edge can be configured for each time measurement function. When a configured starting edge occurs, the value of the internal timer is saved in a FIFO. This FIFO holds up to 16 elements. When the actual trigger edge occurs, the difference in time between the starting edge and the triggered edge is copied to the respective register.
Bits 8 to 11 "Previous start edge" of the "CfO_EdgeTimeFallingMode" on page 42 and "CfO_EdgeTimeRisingMode" on page 43 registers can be used to define which detected starting edge from the FIFO should be used to calculate the difference. Additionally, when the trigger edge occurs, the counter clocked internally using bits 12 to 15 "Time measurement resolution" are copied to the "TimeStampFallingCH" on page 45 and "TimeStampRisingCH" on page 45 registers.

### Information:

**The time measurement function is an extension of edge detection, so all of the channels used must be configured there.**

### 14.10.1 Enable time measurement function

Name:
CfO_EdgeTimeglobalenable

This register enables/disables the time measurement function for the entire module.

| Data type | Values |
|---|---|
| USINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | Time measurement function | 0 | Disabled for entire module |
| | | 1 | Enabled for entire module |
| 1 - 7 | Reserved | - | |

### 14.10.2 Configure time measurement function for the falling edge

Name:
CfO_EdgeTimeFallingMode01 to CfO_EdgeTimeFallingMode15

These registers can be used to configure the time measurement function for the falling edge of the respective channel.

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 - 3 | Selects the channel for the starting edge | 0 | Channel 1 |
| | | ... | |
| | | 14 | Channel 15 |
| 4 | Selects the edge for the starting edge | 0 | The falling edge of the channel configured in bits 0 to 3 serves as the starting edge. |
| | | 1 | The rising edge of the channel configured in bits 0 to 3 serves as the starting edge. |
| 5 - 6 | Reserved | - | |
| 7 | Trigger | 0 | Triggered[1)] |
| | | 1 | Continuous[2)] |
| 8 - 11 | Previous start edge | 0 to 15 | The value determines which entry in the starting edge FIFO should be used to calculate the time difference. |
| 12 - 15 | Time measurement resolution | 0 | 8 MHz |
| | | 1 | 4 MHz |
| | | 2 | 2 MHz |
| | | 3 | 1 MHz |
| | | 4 | 500 kHz |
| | | 5 | 250 kHz |
| | | 6 | 125 kHz |
| | | 7 | 62.5 kHz |

1) The time measurement is triggered by the corresponding bit in register "TriggerFallingCH" on page 43.
2) Time measurement runs continuously and is triggered at every edge.

### 14.10.3 Configure time measurement function for the rising edge

Name:
CfO_EdgeTimeRisingMode01 to CfO_EdgeTimeRisingMode15

These registers can be used to configure the time measurement function for the rising edge of the respective channel.

| Data type | Values |
|---|---|
| UINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 - 3 | Selects the channel for the starting edge | 0 | Channel 1 |
| | | ... | |
| | | 14 | Channel 15 |
| 4 | Selects the edge for the starting edge | 0 | The falling edge of the channel configured in bits 0 to 3 serves as the starting edge. |
| | | 1 | The rising edge of the channel configured in bits 0 to 3 serves as the starting edge. |
| 5 - 6 | Reserved | - | |
| 7 | Trigger | 0 | Triggered[1)] |
| | | 1 | Continuous[2)] |
| 8 - 11 | Previous start edge | 0 to 15 | The value determines which entry in the starting edge FIFO should be used to calculate the time difference. |
| 12 - 15 | Time measurement resolution | 0 | 8 MHz |
| | | 1 | 4 MHz |
| | | 2 | 2 MHz |
| | | 3 | 1 MHz |
| | | 4 | 500 kHz |
| | | 5 | 250 kHz |
| | | 6 | 125 kHz |
| | | 7 | 62.5 kHz |

1) The time measurement is triggered by the corresponding bit in the "TriggerRisingCH" on page 44 register.
2) Time measurement runs continuously and is triggered at every edge.

### 14.10.4 Trigger falling edge detection

Name:
**1:** TriggerFallingCH01 to TriggerFallingCH08
**2:** TriggerFallingCH09 to TriggerFallingCH15

If bit 7 "Trigger" is cleared in the "CfO_EdgeTimeFallingMode" on page 42 register, then detection of a falling edge on the respective input can be triggered using the respective bit in this register. After a bit has been set, the next falling edge on the corresponding channel is detected.

| Data type | Values |
|---|---|
| USINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | 1: TriggerFallingCH01<br>2: TriggerFallingCH09 | 0 | Falling edges on the channel are not detected. |
| | | 1 | The next falling edge on the channel will be detected. |
| ... | | ... | |
| 6 | 1: TriggerFallingCH07<br>2: TriggerFallingCH15 | 0 | Falling edges on the channel are not detected. |
| | | 1 | The next falling edge on the channel will be detected. |
| 7 | 1: TriggerFallingCH08<br>2: Reserved | 0 | Falling edges on the channel are not detected. |
| | | 1 | The next falling edge on the channel will be detected. |

### 14.10.5 Trigger rising edge detection

Name:
**1:** TriggerRisingCH01 to TriggerRisingCH08
**2:** TriggerRisingCH09 to TriggerRisingCH15

If bit "Continued/Triggered" in register "CfO_EdgeTimeRisingMode" on page 43 is cleared, then detection of a rising edge on the respective input can be triggered using the respective bit in this register. After a bit has been set, the next rising edge on the corresponding channel is detected.

| Data type | Values |
|---|---|
| USINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | 1: Trigger rising edge - Channel 01<br>2: Trigger rising edge - Channel 09 | 0 | Rising edges on the channel are not detected. |
| | | 1 | The next rising edge on the channel will be detected. |
| ... | | - | |
| 6 | 1: Trigger rising edge - Channel 07<br>2: Trigger rising edge - Channel 15 | 0 | Rising edges on the channel are not detected. |
| | | 1 | The next rising edge on the channel will be detected. |
| 7 | 1: Trigger rising edge - Channel 08<br>2: Reserved | 0 | Rising edges on the channel are not detected. |
| | | 1 | The next rising edge on the channel will be detected. |

### 14.10.6 Show first falling trigger edge

Name:
**1:** BusyTriggerFallingCH01 to BusyTriggerFallingCH08
**2:** BusyTriggerFallingCH09 to BusyTriggerFallingCH15

If edges are triggered via the bits in the "TriggerFallingCH" on page 43 register, then a set bit in this register indicates that no falling edges have been detected on the respective channel since the corresponding bit was set in the "TriggerFallingCH" register. If a falling edge occurs on the respective channel, then the corresponding BusyTriggerFalling bit is cleared.

| Data type | Values |
|---|---|
| USINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | 1: BusyTriggerFallingCH01<br>2: BusyTriggerFallingCH01 | 0 | A falling edge was detected on the channel. |
| | | 1 | The module is waiting for a falling edge on the channel. |
| ... | | ... | |
| | 1: BusyTriggerFallingCH07<br>2: BusyTriggerFallingCH15 | 0 | A falling edge was detected on the channel. |
| | | 1 | The module is waiting for a falling edge on the channel. |
| 7 | 1: BusyTriggerFallingCH08<br>2: Reserved | 0 | A falling edge was detected on the channel. |
| | | 1 | The module is waiting for a falling edge on the channel. |

### 14.10.7 Show first rising trigger edge

Name:
**1:** BusyTriggerRisingCH01 to BusyTriggerRisingCH08
**2:** BusyTriggerRisingCH09 to BusyTriggerRisingCH15

If edges are triggered via the bits in the "TriggerRisingCH" on page 44 register, then a set bit in this register indicates that no rising edges have been detected on the respective channel since the corresponding bit was set in the "TriggerRisingCH" register. If a rising edge occurs on the respective channel, then the corresponding BusyTriggerRising bit is cleared.

| Data type | Values |
|---|---|
| USINT | See the bit structure. |

Bit structure:

| Bit | Description | Value | Information |
|---|---|---|---|
| 0 | 1: BusyTriggerRisingCH01<br>2: BusyTriggerRisingCH09 | 0 | A rising edge was detected on the channel. |
| | | 1 | The module is waiting for a rising edge on the channel. |
| ... | | ... | |
| 6 | 1: BusyTriggerRisingCH07<br>2: BusyTriggerRisingCH15 | 0 | A rising edge was detected on the channel. |
| | | 1 | The module is waiting for a rising edge on the channel. |
| 7 | 1: BusyTriggerRisingCH08<br>2: Reserved | 0 | A rising edge was detected on the channel. |
| | | 1 | The module is waiting for a rising edge on the channel. |

### 14.10.8 Count falling trigger edges

Name:
CountFallingCH01 to CountFallingCH15

These registers contain cyclic counters that are incremented with every detected falling edge on the respective channel.

| Data type | Value | Information |
|---|---|---|
| USINT | 0 to 255 | Counter for falling edges |

### 14.10.9 Count rising trigger edges

Name:
CountRisingCH01 to CountRisingCH15

These registers contain cyclic counters that are incremented with every detected rising edge on the respective channel.

| Data type | Value | Information |
|---|---|---|
| USINT | 0 to 255 | Counter for rising edges |

### 14.10.10 Timestamp of falling edge

Name:
TimeStampFallingCH01 to TimeStampFallingCH15

When a falling edge occurs on the respective channel, the current counter value of the module timer is copied to these registers.

| Data type | Value | Information |
|---|---|---|
| UINT | 0 to 65535 | Timestamp for rising edges |

### 14.10.11 Timestamp of the rising edge

Name:
TimeStampRisingCH01 to TimeStampRisingCH15

When a rising edge occurs on the respective channel, the current counter value of the module timer is copied to these registers.

| Data type | Value | Information |
|---|---|---|
| UINT | 0 to 65535 | Timestamp for rising edges |

### 14.10.12 Time difference of falling edge

Name:
TimeDiffFallingCH01 to TimeDiffFallingCH15

When a falling edge occurs on the respective channel, the time difference compared to the starting edge configured in bit 4 of the "CfO_EdgeTimeFallingMode" on page 42 register is copied to this register.

| Data type | Value | Information |
|---|---|---|
| UINT | 0 to 65535 | Time difference from starting edge |

### 14.10.13 Time difference of rising edge

Name:
TimeDiffRisingCH01 to TimeDiffRisingCH15

When a rising edge occurs on the respective channel, the time difference compared to the starting edge configured in bit 4 of the "CfO_EdgeTimeRisingMode" on page 43 register is copied to this register.

| Data type | Value | Information |
|---|---|---|
| UINT | 0 to 65535 | Time difference from starting edge |

## 14.11 Minimum cycle time

The minimum cycle time specifies the time up to which the bus cycle can be reduced without communication errors occurring. It is important to note that very fast cycles reduce the idle time available for handling monitoring, diagnostics and acyclic commands.

| Minimum cycle time |
|---|
| 150 µs |

## 14.12 Minimum I/O update time

The minimum I/O update time defines how far the bus cycle can be reduced while still allowing an I/O update to take place in each cycle.

| Minimum I/O update time |
|---|
| 150 µs |