White Paper

Safe FPGA Encoder Solution

Multi-Protocol Position Encoder Interface IP for Safety-Related Automation Drives

Integration of Multi-Protocol Position Encoder Interface IP cores in Intel's Edge-Centric FPGAs for various automation drives.

Authors

Jens Onno Krah
Prof. Dr.-Ing.
Technische Hochschule Köln

Tobias Schmidt
M.Sc.
Technische Hochschule Köln

Rolf Richter
Dipl. Ing.
Intel Programmable Solutions Group

Table of Contents

Multi-Protocol Position Encoder Interface IP for Safety-Related Automation Drives

The manufacturers of servo drives have the challenge to support different position encoder interfaces. Migration to 2-wire interfaces is simplifying support for these encoders. Data transfers generally occur over RS-485 transceivers at 10 to 25 Mbits/sec using either 8b/10b coding or Manchester coding to allow a DC-free half-duplex transmission without an extra clock signal. The position encoders (usually for motor integration) can thus be supplied with power via the same two wires.

The amount of on-chip memory in FPGAs has greatly increased over the years, so we developed a universal encoder interface for FPGAs to use the available on-chip dual-port RAM resources as a common protocol-independent encoder interface for servo control including Safe Logic. This design enables particularly simple integration of different encoder protocols in automation drives. Connection to a higher-level central safety controller for multi-axis applications is also easy to implement and it accommodates multiple safety-related encoder protocols in one system.

A safety communication layer (SCL) uses additional measures to ensure safe transmission of data in accordance with the requirements of IEC 61508. An SCL can be built with a black channel, a defined communication system that contains one or more elements without evidence of design or validation according to IEC 61508 [1].

The multi-protocol position encoder interface intellectual property (IP) is part of a black channel, which means that the IP core does not need to be examined from a safety point of view. This significantly simplifies the system integration of the following three fully digital encoder interfaces:

- EnDat 3
- Single Cable Solution (SCS) open link
- HIPERFACE DSL

The multi-protocol position encoder interface IP can be integrated to Intel® MAX® and Intel® Cyclone® FPGAs such as Intel® MAX® 10, Cyclone® V, and Intel® Cyclone® 10 FPGAs.

Introduction

It is common in servo drives to use a combination of microcontroller (µC) and FPGA (usually from different vendors) for high-performance motor control. Modern encoder protocols, e.g. EnDat 3, and innovative algorithms, e.g. current observers, are not yet supported by µC devices. A further disadvantage of such highly specialized µC devices is that these µCs do not have a second source.

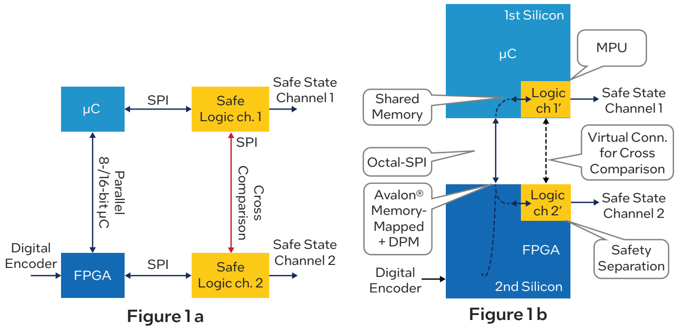

Figure 1. Typical control configuration for automation drives
a) Traditional approach with two additional µCs for the Safe Logic
b) Modern approach with one µC and one FPGA

This paper describes how different modern position encoder protocols (provided as different master IP cores) can be integrated into Intel® FPGAs. Instead of a complex parallel bus interface or a slow Serial Peripheral Interface (SPI), a quad or octal SPI is used for the communication with the µC, which is already common for the connection of flash memories. Inside the FPGA, the encoder protocol master IP cores are connected via the Avalon® memory-mapped interface bus configured by the Platform Designer.

This is particularly advantageous if an automation drive is also designed with safety features. A common approach is to use two independent safety-channel structures, as an example, a 1st channel can be implemented through the µC and the 2nd through the FPGA.

Figure 1a shows the typical control architecture for automation drives. With the traditional approach, a µC with an FPGA is used for non-safe control. Two additional small µC devices are provided for the 2-channel Safe Logic.

Because the safety-related µC devices usually don't require the highest performance, the algorithms of these controllers are taken over by the originally non-safe controller and the FPGA in a new approach shown in Figure 1b. The Safe Logic in the FPGA can be implemented in VHDL or with the help of a softcore processor, e.g. Nios® II processor. However, the Safe Logic has to be isolated from non-safe motor control:

- For the µC, available features can be used for safety-related separation. For example, dedicated hardware like hardware virtualization or a memory protection unit (MPU) can support the software isolation required for functional safety. Shared memory regions can be used for very fast black channel communication between safe and non-safe sections [1]. Approximately 5% of the µC's processing time should be reserved for the Safe Logic.
- For the FPGA, a safety separation design flow based on logic isolation can be used. On-chip dual-port memory allows very fast black-channel communication between safe and non-safe sections.

Closing the Servo Loop with an Encoder

To close the servo control loops, the sampled position value δ_k has to be transmitted with a high resolution, high update rate, and low latency. Figure 2 shows a basic schematic of modern encoders with a 2-wire interface. Typical are 40- to 48-bit resolution and a cycle time of 62.5 µs. Sampling, coding, and transmission cause a total delay time of approximately 25 µs, depending on the encoder protocol used.

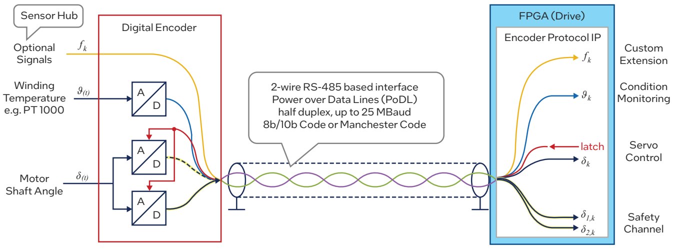

Figure 2. Basic schematic of modern encoders with a 2-wire interface.

In multi-axis systems with central control, the synchronization of the control loops and the position sampling times is very important. The clock master is usually a higher-level controller, for example, an industrial PC used to control a CNC machine. Each connected drive uses an integrated phase-locked loop (PLL) to synchronize the Pulse Width Modulation (PWM) carrier signal with the centralized control. For example when using EtherCAT, which is widely used in motion control systems, its sync event is available for this purpose. The encoder latch signal for the encoder protocol IP can be derived from the PWM carrier signal.

For safe motion functions according to IEC 13849 Cat. 3 or Cat. 4, a safety-related position is also transmitted with two independent channels using the black-channel principle. Even though safety-related multi-turn encoders have not been common for motor integration so far, the safety protocols are prepared for them. Typical safety-related loop cycle times start from 250 µs.

It is usual to transmit additional data such as temperature with lower priority. If the winding temperature of the motor is measured with the widely used PT 1000 sensor, it is sufficient to transmit the digitized value ϑ_k at a lower update rate – e.g. every 5 ms. This is usually done by a multiplex algorithm via a service data channel.

Many transmission protocols allow optional digital data (e.g. from other sensors) to be transmitted cyclically. For this purpose, these digital data words f_k can be integrated into the individual protocols in each case.

Figure 3 shows the logical signal flow of encoder-related signal processing. The high-resolution, non-safe position is very time-critical and is usually processed with a µC using an interrupt service routine (ISR). Alternatively, an FPGA can perform this processing faster with a parallel processing algorithm. The encoder latch signal (usually generated by programmable logic with negligible jitter) is not shown in detail. High sampling rates in combination with high resolution and low signal delay enable high speed-loop control bandwidths.

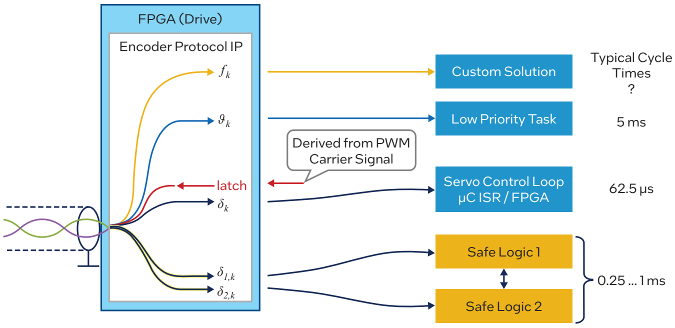

Figure 3. Logical signal flow of encoder-related signal processing.

From the µC's point of view, the position data should be automatically sampled and transmitted synchronously with the control loop cycle time. Retrieving the already sampled actual position then requires only a read access from an FPGA internal register.

The IP block inside the FPGA automatically starts the sample and transmission of the position value synchronized to the control loops. The µC just has to read the data from the FPGA with no additional overhead.

Isolated from external influences, Safe Logic channel 1 and Safe Logic channel 2 process the safety-related algorithms with redundancy; hardware fault tolerance (HFT) = 1. For safety-related motion monitoring, it is usually not advantageous to sample overly fast. In the numerical calculation of the speed, short sampling times cause higher signal noise. Additional low-pass filters for noise suppression add signal delay. Therefore, an update cycle time of 0.25 to 1 ms is typically sufficient for safe motion.

Physical Architectures for Connection

Figure 4 shows one physical architecture option for encoder-related signal processing. The main controller communicates with the FPGA using either a (16- or 8-bit) parallel µC interface or a multi-bit SPI (Quad / Octal). The two µC devices implementing the Safe Logic each use an independent SPI.

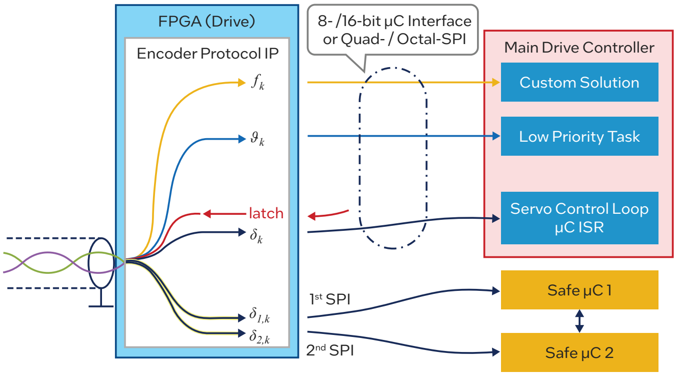

Figure 4. Physical architecture for encoder-related signal processing: 1st option.

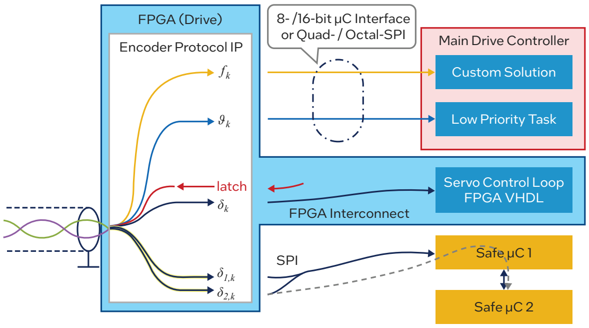

Figure 5. Physical architecture for encoder-related signal processing: 2nd option.

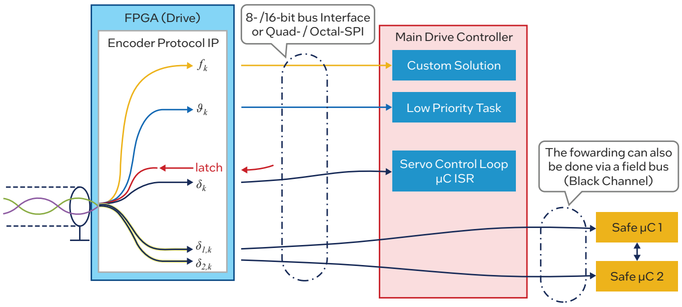

Figure 6. Physical architecture for encoder-related signal processing: 3rd option.

Figure 5 shows an architecture with parallel algorithm processing within the FPGA for faster processing times. The 40- or 48-bit wide position value is then routed directly within the FPGA.

Only the first safety µC has a direct SPI connection to the FPGA and forwards the safety-related encoder data of the 2nd channel to the 2nd safety µC.

A third particularly flexible architecture is shown in Figure 6. The main drive controller forwards the safety-related protocol data units (PDU) to the centralized Safe Logic for multiple drives. The non-safe controller is then part of the black channel of the safety-related encoder PDU.

Due to space limitations, not all possible interface combinations are presented. For example, with a hypervisor for isolation, the main drive controller can be split into two virtual machines (VM). One safety-related VM can be used for the first Safe Logic block and would either access the safety-related encoder data directly via the memory interface, or it could receive it via a shared memory area. In terms of encoder IP connectivity, this architecture would be identical to the third option.

Quad or Octal SPI

For a long time, memory and peripherals were connected to µC devices either using a comparatively slow SPI with typical data rates of 10 Mbit/s. Alternatively, a parallel SRAM interface might achieve an access time somewhat less than 100 ns. However, a classical parallel bus interface with e.g. a 16-bit data bus, a 20-bit address bus, and the associated control signals needs almost 40 connections between the µC and its periphery.

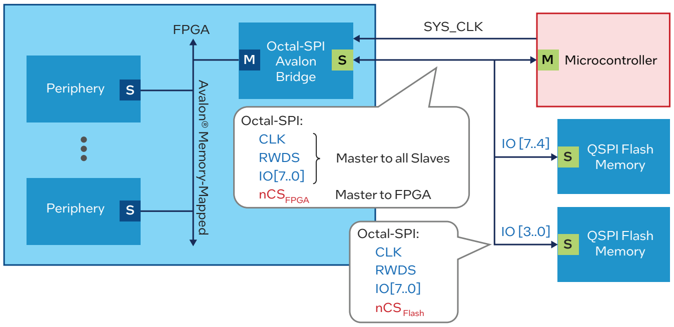

Figure 7. Quad and Octal SPI are enabling high data rates between a µC and its periphery with only a low number of connections.

Nowadays, it is common practice to use Quad SPI interfaces that employ four bidirectional data lines to connect flash memories with µC devices. Due to clock rates of up to 100 MHz and 4-bit/clock transfers, significantly higher data rates are possible compared with parallel or SPI connections [2]. Many µC devices already support octal SPI connections with twice the bandwidth, formed by connecting two Quad SPI ports in parallel.

Quad or Octal SPI ports can also connect FPGAs very efficiently with powerful µCs with very high bandwidths using only a few connections.

Figure 7 shows an FPGA and two flash memories connected to a µC via an Octal SPI. Here, the µC acts as host with an Octal SPI master. The FPGA and the flash memories are Octal SPI slaves. Within the FPGA, an Octal SPI Slave to Avalon Master bridge provides a connection between the host and the remote system for Octal SPI transactions. The periphery connected to the Avalon interface bus can be configured with the Platform Designer as known from the softcore CPU Nios II processor [3].

Typically, today's µCs for drives are equipped with 32-bit architectures. If the µC performs a 32-bit wide read access to an 8-bit wide peripheral component, this access is automatically divided into four sequential byte accesses by the bus system. The µC has to wait until the data word is fully read. A drawback of the Quad and Octal SPI technology is that additional wait cycles cannot be requested from slower peripherals.

To obtain a universally usable and flexibly expandable approach, it makes sense to configure a 32-bit wide Avalon Master interface. 8-, 16- and 32-bit wide data words are thus always read or written consistently in one clock cycle.

Although flash memories connected via Quad SPI do not require an additional clock, it is advantageous for an FPGA slave connection if an associated synchronous clock is available.

Even though the data is transferred nibble- or byte-wise, care must still be taken to ensure that the byte arrangement fits. The reference design for the Encoder IP is configured for µC devices that use the little-endian format.

2-wire Encoder Interfaces

One advantage of the RS-485 2-wire encoder interface technology is that no additional feedback cable is required. The encoder receives power (e.g. 12 V) via the same 2-wire interface. Data is transmitted serially, half-duplex with up to 25 MBaud. 8b/10b or Manchester coding ensures a zero DC signal offset and allows clock recovery.

In the cable for the power connection of the servo motor, the two signal wires (2-wire) that were previously used to connect a thermal sensor are now used for the digital encoder interface. These two signal lines should be twisted and provided with a separate shield. The used terminating resistor at the RS485 transceiver has to match the characteristic impedance of the cable. A characteristic impedance of approximately 110 Ω is typical.

EnDat 3

The EnDat 3 interface from HEIDENHAIN works with fixed transmission rates of 12.5 or 25 Mbit/s (effective) and Manchester code is used to generate an implicit clock with no DC offset. The RS-485 transceiver must be capable of handling 25 or 50 Mbits/s respectively. The EnDat 3 Master IP core requires a clock frequency of at least 100 MHz, resulting in an 8X sample rate for Manchester-encoded bits at 12.5 Mbit/s [4]. EnDat 3 is certified up to SIL3 and the communication link including the master IP core is part of the black channel.

The physical interface uses four or two wires, depending on whether the power supply and the data for the encoder are transmitted separately or is modulated onto the power lines. EnDat 3 offers the advantage that real-time process data can be pre-configured for multiplexing with send lists – an EnDat 3 specific process data multiplex setup – to off-load the interrupt service routine for control in the drive.

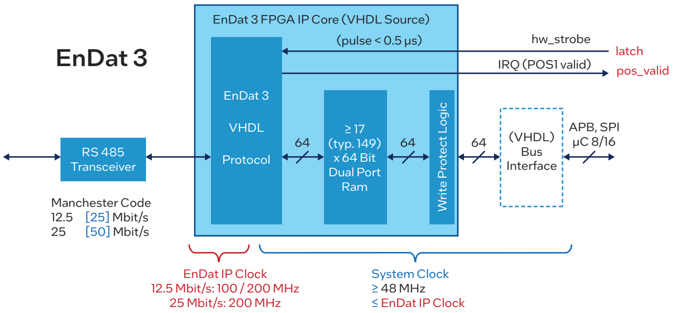

Figure 8. The EnDat 3 Master IP core from HEIDENHAIN is based on a 64-bit dual-port RAM-based register array [4].

The IP core divides EnDat 3 communications into two interfaces:

1. Routable signals including the mandatory cyclical input signal **hw_strobe** for high precision latching of the encoder position usually divided from the PWM carrier signal. Up to four further events are configurable. They can be used for an ISR but can also be routed for FPGA-based algorithm processing. Figure 8 shows the configured event **POS1_valid**, which indicates that the non-safe position has been received.

2. A response register set with a configurable number of 64-bit registers. At least 17 response registers are necessary for communication. Typical are configurations with 149 registers to buffer various messages in their associated response registers. This dual-port RAM-based interface is used for configuration, to read the encoder position, for non-safe and safety-related messages, and so on.

The basic 64-bit wide dual-port RAM interface of the EnDat Master IP core enables very flexible connections. It provides a very high data throughput rate, which is especially advantageous when the data is processed internally by the FPGA. Its system clock (chosen according to the drive application) has to be greater than 48 MHz and also less than the EnDat 3 Master IP core clock frequency.

The basic 64-bit dual-port RAM interface for the HEIDENHAIN encoder interface provides the following interfaces to connect with the drive µC:

- Advanced Peripheral Bus (APB), which is designed for on-chip access to system peripherals
- Standard SPI with a clock frequency up to one-eighth of the system clock
- Parallel µC interface (8- or 16-bit).

Figure 9 shows how the EnDat 3 Master IP core from HEIDENHAIN is internally connected to an Octal-SPI to Avalon bridge within the FPGA. The 32-bit wide Avalon memory-mapped interface bus can also be used by other FPGA internal peripherals. During read access to the 64-bit EnDat 3 dual-port RAM, the associated 64-bit wide content is latched in the EnDat 3 Master IP 32-bit bus adapter when the lower 32 bits are read. This stored content is read out when the upper 32 bits are accessed. Therefore, no inconsistent data is read during 64-bit register accesses. A preferred configuration is when the system clock is exactly twice as fast as the Octal SPI clock, e.g. 66 / 33 MHz.

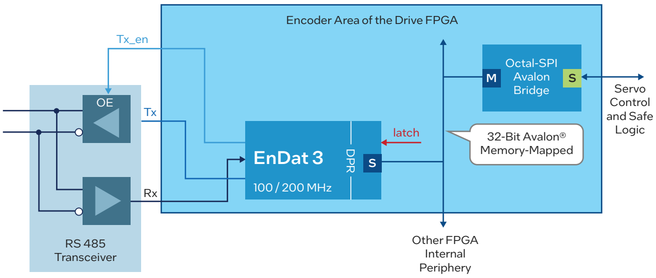

Figure 9. The EnDat 3 IP core from HEIDENHAIN is connected with the drive controller using an Octal SPI to Avalon bridge.

This configuration fits the third physical architecture shown in Figure 6, where the Safe Logic is connected via a µC.

Figure 10 shows an extended configuration with on-chip RAM and a programmable Direct Memory Access (DMA) unit. Similar to how an EtherCAT Master assembles an Ethernet frame with many output data words, this DMA unit can quickly form a contiguous data frame in the on-chip RAM with preconfigured input data in an event-driven manner. The on-chip RAM buffering allows the higher-level controller to read the data blocks via the Octal SPI in burst-mode with fewer accesses and greater speed. In the same way, for example, the position can be provided via the Avalon interface bus for data processing within the FPGA.

With the DMA as second Avalon interface bus master, DMA accesses can only occur if the Octal SPI does not have read access, e.g. cyclically, shortly after the encoder IP data has been updated.

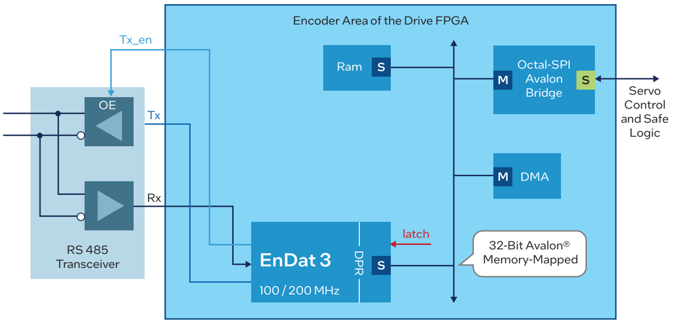

Figure 10. The EnDat 3 Master IP core with additional on-chip RAM and a programmable DMA unit.

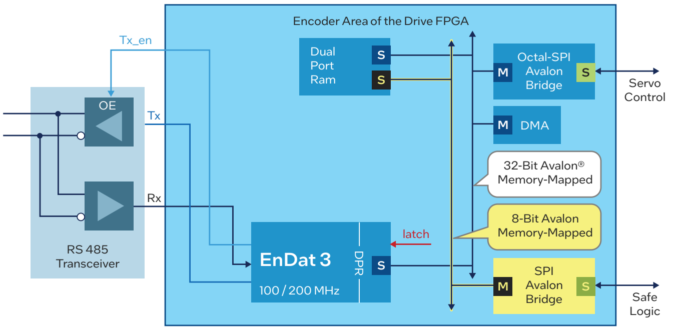

Figure 11. The EnDat 3 Master IP core with dual-port RAM and a separate safety SPI with its own 8-bit Avalon interface bus.

This means that access to the dual-port RAM via the Octal SPI port must be synchronized so that it cannot occur during the data update and for a short period thereafter.

The Safe Logic can be connected to the FPGA via a separate SPI port using the existing Avalon memory-mapped interface bus with an additional SPI to Avalon bridge.

However, it is more flexible if the Safety SPI is equipped with its own Avalon memory-mapped interface bus as shown in Figure 11. In this case, an approach with 8-bit data width is sufficient.

The EnDat 3 IP core, provided by Heidenhain, is coded in VHDL. The source code is free of charge but requires a signed contract with Heidenhain. Therefore, the reference design uses encrypted VHDL files, which can easily be replaced by the original files.

SCS open link

SCS open link is an open and standardized interface for motor feedback systems from Hengstler. SCS open link 2.0 is the successor of version 1.0, which was formally called the Acuro link interface. The SCS implementation can be realized with 2- and 4-wire connections, which are compatible with both EnDat 3 interfaces mentioned above and use RS-485 transceivers for data transmission. In addition to the original versions implemented with FPGAs, the SCS open link protocol can also be implemented as master and slave using standard µC devices equipped with 10 Mbit/s UARTs [5]. The relatively low transmission frequency of 10 Mbits/s as well as additional start and stop bits at byte level and the 8b/10b coding results in a higher level of interference immunity. The effective data rate is 6.4 Mbit/s. The safety-related transmission relies on black channel technology and is certified up to SIL3. The exchange of data and information is controlled by commands. Like EnDat 3, SCS open link has a single-cycle architecture and therefore enables the exchange of all data (e.g. position, safety telegram, OEM-RD / -WR, or other process and condition monitoring data) within a single cycle.

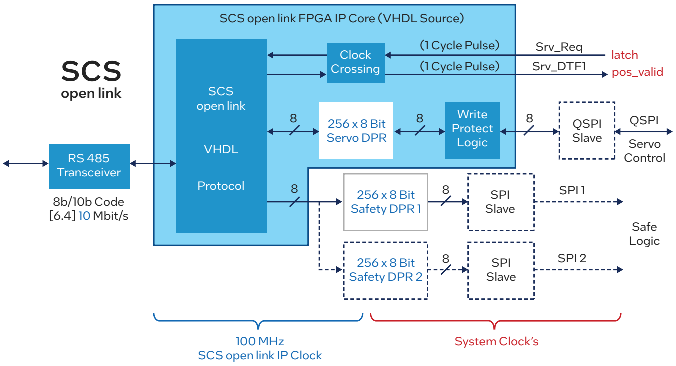

Figure 12. The SCS open link Master IP core comes with 8-bit DPR interfaces: one for servo control and one or two for the Safe Logic [5].

The SCS open link Master IP core requires a clock frequency of exactly 100 MHz. The 8-bit wide dual-port RAM interfaces (one for servo control and one for safety) permit easy clock-crossing and short access times within the FPGA. Figure 12 shows a simplified block diagram. For the safety interfaces, the dual-port RAM is not integrated into the IP core. This allows a designer to optionally instantiate two dual-port RAM blocks, each with its SPI port. The dual-port RAM interfaces are organized as banks with 256 8-bit registers, for reading out the actual values. The servo dual-port RAM can also be used for configuration.

The IP divides the SCS open link communication into five logical interfaces:

1. IP core signals ready for routing within the FPGA. The cyclical input signal (one cycle pulse) **Srv_Req** for high precision latching of the encoder position is mandatory. Also shown is the optional output signal **Srv_DTF1** which indicates that the non-safe position is updated.
2. Servo dual-port RAM (DPR)
   The up to 40-bit (non-safe) process data – typical 16-bit multi-turn and 24-bit single turn – can be read with accesses of up to 5 bytes from the servo dual-port RAM. It is updated with the servo update cycle initiated with Srv_Req, and also IP core internally initiated from the safety partition within the dual-port RAM with the safety update cycle.
3. Diagnostics and flash memory access, e.g. an electronic type plate for the motor specified in JSON, the measured and digitized winding temperature.
4. Sensor hub data, e.g. the measured and digitized vibration or torque information.
5. Safety DPR 1 and optional safety DPR 2:
   1st and 2nd safety-related position.

Figure 13 shows the integration of the SCS open link Master IP core. Three major options are possible:

1. The Servo DPR and the Safety DPR are both connected to the 32-bit Avalon memory-mapped interface bus. The Octal SPI Avalon bridge is used for all connections.
2. The Servo DPR and the Safety DPR 1 are both connected to the 32-bit Avalon memory-mapped interface bus. At least one channel of the Safe Logic uses an additional SPI for independent reading the safety data.
3. Servo DPR is connected to the 32-bit Avalon memory-mapped interface bus. The Safe Logic uses two independent SPI ports for reading the safety-related data.

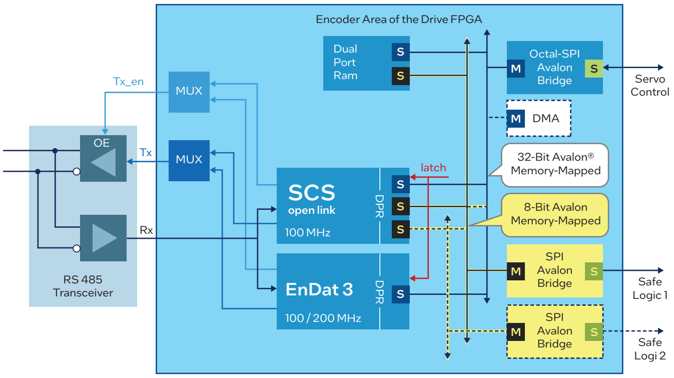

Figure 13. The EnDat 3 and SCS open link IP cores for connection with an Octal SPI and up to two separate safety SPIs with their own 8-bit Avalon interface buses.

The Servo dual-port RAM Avalon interface connection uses only the lower 8 bits of the Avalon interface data bus but reads and writes to this dual-port RAM employ full 32-bit accesses. Data bits D31 down to D8 are not used. Because the 32-bit accesses are not split into four sequential 8-bit accesses, the data is immediately available. The Octal SPI Avalon bridge can be activated in such a way that it only transfers the requested data (least significant byte) without access-related delay. This also makes word or double-word alignments within the 8-bit DPR unnecessary.

The programmable DMA device is not required for the SCS open link Master IP core, but it can be used to assemble data blocks for continuous reads with the Octal SPI in burst mode.

The SCS open link IP core used, which is provided by Hengstler, is coded in VHDL. The source code is free of charge but requires a signed contract with Hengstler. Therefore, the reference design uses encrypted VHDL files, which can easily be replaced by the original files.

HIPERFACE DSL

HIPERFACE Digital Servo Link (HIPERFACE DSL) from SICK was the first fully digital encoder protocol for motor-integrated feedback systems using the 2-wire interface. The optional safety-related transmission is certified up to SIL 3.

Figure 14 shows a simplified block diagram of the HIPERFACE DSL Master IP core from SICK [6]. The left side of the figure shows the 2-wire interface to the HIPERFACE DSL encoder via an RS-485 transceiver. The right side interfaces to the superimposed controllers – servo control and Safe Logic. For HIPERFACE DSL, the Master IP core clock is always 75 MHz (± 100 ppm). The HIPERFACE DSL transmission bit rate of 9.375 MHz is one-eighth of this IP clock rate. Due to the 8b/10b coding, the effective data rate is 7.5 Mbit/s. The Master IP core itself is delivered as encrypted HDL code. For compilation, a license file from SICK is required.

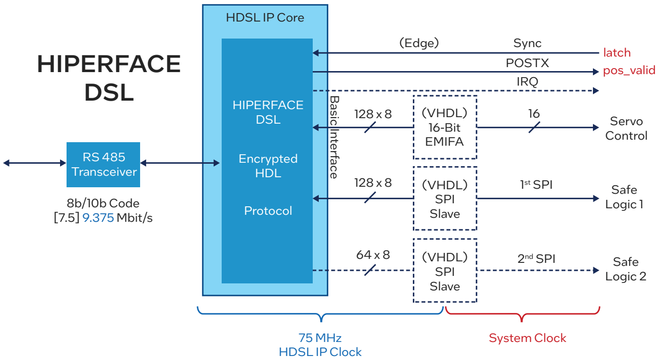

Figure 14. HIPERFACE DSL Master IP core from SICK [6].

The HIPERFACE DSL communication is divided into up to five logical interfaces:

1. HIPERFACE DSL Master IP core signals, ready for routing within the FPGA. The cyclical input signal **sync** for high precision latching of the encoder position is required. This signal is usually derived from the PWM carrier signal. Also shown are the optional output signals **POSTX** and **IRQ**.

2. The **drive interface** forms the essential communication interface between the non-safe drive control and the HIPERFACE DSL encoder. The absolute position data and the configuration of the motor feedback system are available via this interface. The communication is structured as an array of 128 8-bit wide registers for memory-mapped read and write access within the HIPERFACE DSL clock domain. This register array is compiled into logic elements and does not use on-chip dual-port RAM. There are two options for interfacing to these registers:

    I. An SPI block – provided with the IP core by SICK as VHDL source code – can be used to connect to a superimposed drive µC. The maximum specified clock rate for this SPI port is 10 MHz.

    II. A 16-bit parallel interface block specified in VHDL source code is also supplied with the IP core.

3. **Safe 1 interface** is used to connect the safety part of the motor feedback system for configuration and for reading a first (primary) safety-related position. This interface is structured as an array of 128 8-bit wide registers for memory-mapped read and write access. SICK recommends using the supplied SPI interface block to connect with the first Safe Logic.

4. **Safe 2 interface** is used to connect the safety part of the motor feedback system for reading a secondary safety-related position. It is structured as an array of 64 8-bit wide registers for memory-mapped read-only access. SICK recommends using the supplied SPI interface block for connection with the second Safe Logic.

5. Not shown in the simplified figure is the optional **SPI PIPE interface**. SPI PIPE is a read-only interface for sensor hub data. Alternatively, the sensor hub data can be read via the **drive interface**.

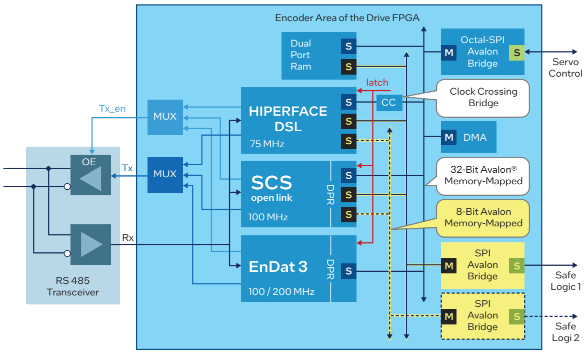

Figure 15. Multi encoder Master IP core interface.

Fig. 15 shows the multi-encoder interface with three different encoder protocol Master IP cores. Since the HIPERFACE DSL IP core does not use on-chip dual-port RAM, an Avalon interface clock-crossing bridge is implemented. In addition, a distinction must be made between read and write accesses.

- Reading from the HIPERFACE DSL Master IP core: The maximum read speed - fully synchronous to the 75 MHz clock is 37.5 MHz due to one wait cycle. The additional delay caused by the clock crossing bridge must also be considered when reading. This is not critical when using DMA, because the DMA controller can employ a wait signal. For an Octal-SPI read burst, the read access speed is not fast enough. Instead, the data can be copied by DMA to an on-chip RAM and the Octal SPI can read the data from the copy in memory.

- Writing to the HIPERFACE DSL Master IP core: The maximum write speed is 9.375 MHz. Therefore, the Octal-SPI burst-mode cannot be used here. Since writing is only required for configuration, this is not a significant restriction.

When using the lower speed standard SPI port, no additional clock synchronization is required.

The HIPERFACE DSL IP core used comes from SICK and is supplied as encrypted HDL code. SICK usually does not provide its source code. Instead, a variety of internal state signals can be evaluated, for example with the Signal Tap II logic analyzer.

Reference Hardware

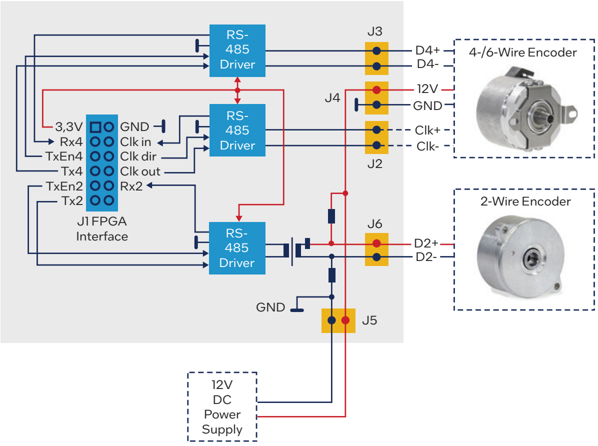

Figure 16. Simplified schematics from an encoder interface with RS 485 drivers.

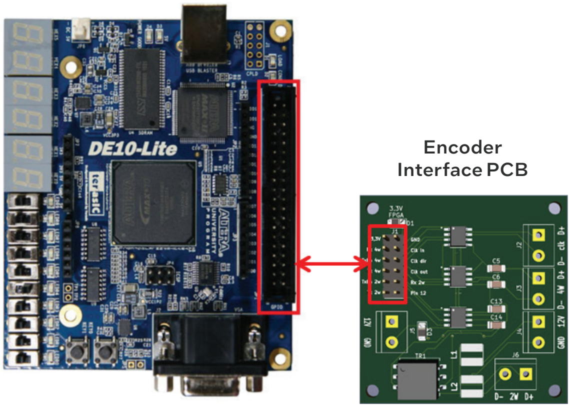

Figure 17. Terasic DE10-Lite from the Intel FPGA University Program with the interface PCB was used for the Encoder IP implementation

Figure 16 shows a simplified schematic of the RS-485 interface board used. The pinout of connector J1 has been designed so that the Terasic DE-10-Lite FPGA board from the Intel FPGA University Program can be connected directly with a standard 12-pin one-to-one ribbon cable, shown in Figure 17. All signals on the J1 connector conform to the low voltage TTL standard. Connector J5 can be used to supply the encoder with an external 12 V DC voltage source. Connector J6 is intended for the connection of 2-wire encoders. 4-wire or classic 6-wire encoders can be operated via J2 to J4.

Schematics, layout, and bill of materials of the interface PCB are available for download on **ResearchGate**.

Deliverables for the Reference Design

- Encrypted IP block
- Documentation with register description
- Encoder Interface PCB production information
- Design example for the Terasic DE10-Lite FPGA board

Conclusion

Various fully digital 2-wire encoders for automation have become established. Corresponding encoder protocol Master IP cores are offered by different manufacturers. RS-485 implementations delivering power over data lines have become well established. However, position encoder interfaces to the drive microcontroller are not standardized and are still based on classic parallel 8- or 16-bit µC interfaces or the simple low-speed SPI standard.

This white paper presents a common Avalon memory-mapped interface bus for EnDat 3, SCS open link, and HIPERFACE DSL. For configuration and the non-safe servo control loop, a fast Octal or Quad SPI interface via the 32-bit Avalon interface bus is exploited.

If the safe logic does not retrieve the safety-related position data via the drive µC, one or two classic SPI ports with an 8-bit Avalon interface bus can be configured. The presented methods can be used via multiple instantiations for multi-axis drives.

References

1. IEC 61784-5-3:2018 Industrial communication networks - Profiles - Part 5-3: Installation of fieldbuses - Installation profiles for CPF 3, August 2018.
2. Quad SPI interface on STM32 microcontrollers and microprocessors, Application note 4760 rev 3, STM, April 2020.
3. Avalon Interface Specifications, Intel, MNL-AVABUSREF, 2021.05.27.
4. EnDat 3 Master IP– Interface specification, HEIDENHAIN, Traunreut, 2019.
5. SCS open link, Master IP Core, User Manual, Hengstler, Aldingen, 2020.
6. HIPERFACE DSL MASTER Safety Integration Manual, SICK, Donaueschingen, Germany, March, 8th 2021.

Intel technologies may require enabled hardware, software or service activation.
No product or component can be absolutely secure.
Your costs and results may vary.
Customer is responsible for safety of the overall system, including compliance with applicable safety-related requirements or standards.