

Matthias Schaffland

## FPGA als Basis

Immer höhere Datenraten gehören zu den Forderungen an die Interface-Technologien in der industriellen Bildverarbeitung. Höhere Datenraten bedeuten aber auch erhöhte Anforderungen an die Hardware, um diese verarbeiten zu können. Hier bieten sich FPGAs als flexible Komponenten an, um die unterschiedlichen Interface-Technologien zu implementieren.

**O**b USB3 Vision, CoaXPress, CLink HS oder GigE Vision – jeder Schnittstellen-Standard in der industriellen Bildverarbeitung hat seine spezifischen Vor- und Nachteile, die bei der Auswahl zu berücksichtigen sind. Gemeinsam ist jedoch allen, dass

sie schnelle serielle Übertragungswege nutzen, die sich mit IO-Ports von Standard-CPU's nicht mehr bedienen lassen. Hierfür bieten sich FPGAs an, die mit den eingebauten Transceivern – je nach Chip – Datenraten von mehr als 20 Gbit/s erreichen können.

Ein weiterer Vorteil des FPGA-Einsatzes: Nicht nur das physikalische Interface kann implementiert werden, sondern auch das zugehörige Protokoll. Die Standards USB3 Vision, CoaXPress, CLink HS oder GigE Vision definieren zum Beispiel den Paketaufbau sowie den Nachrichtenfluss zwischen Device (zum Beispiel Kamera) und einem Host (PC). Zeitkritische Elemente lassen sich in der FPGA-Logik als Hardware realisieren, wohingegen vergleichsweise langsame Komponenten in einem Soft-core-Prozessor in C/C++ implementiert werden können. Dies schafft Flexibilität bei der Realisierung gerätespezifischer Funktionen. Bei geschickter Hardware-Gestaltung lässt sich zudem eine weitestgehend gemeinsame Elektronik zum Beispiel für eine Kamera mit unter-



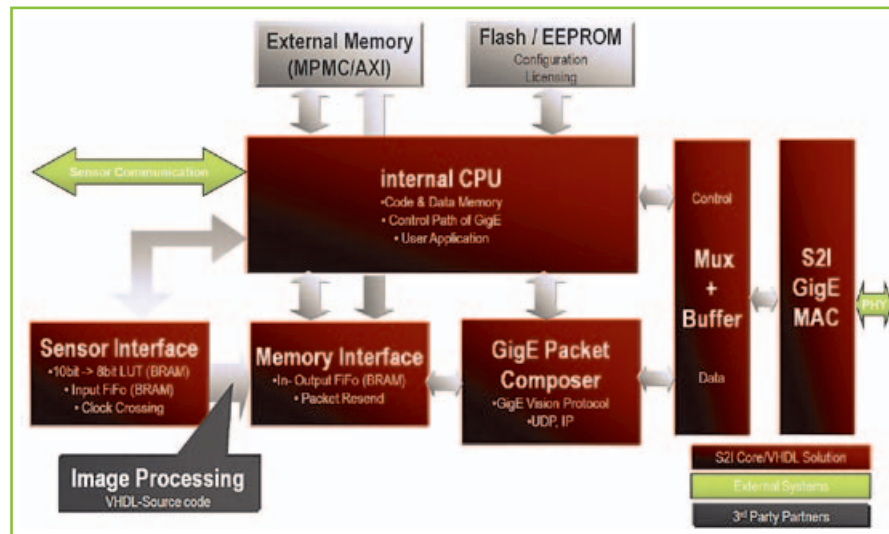
schiedlichen Schnittstellen entwickeln, bei der nur eine Interface-spezifische Leiterplatte unterschiedlich sein muss.

### Die FPGA-Implementierung

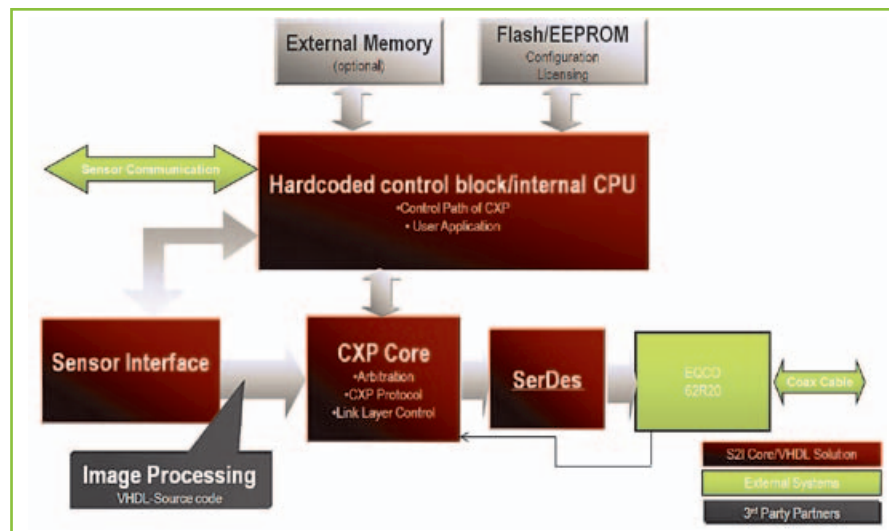
Wie kann nun eine typische FPGA-Implementierung der unterschiedlichen Protokolle aussehen?

### GigE-Vision

Grundsätzlich sind zwei Datenpfade zu unterscheiden: Streaming- beziehungsweise Bilddaten, die von der Kamera zu einem PC übertragen werden, und Control-Daten respektive Kommandos vom Host zur Kamera, um zum Beispiel die Belichtungszeit zu setzen (*Bild 1*). Da der Control-Pfad normalerweise nicht sehr zeitkritisch ist, wird dieser in einer FPGA-integrierten CPU behandelt, beispielsweise einem MicroBlaze oder NIOS. Die CPU lässt sich in C/C++ programmieren und bietet sich auf einfache Weise für Erweiterungen an. Der Streamingpfad jedoch muss hohe Datenraten verarbeiten können, zum Beispiel mehr als 1 GByte/s bei einer 10-Gigabit-Verbindung. Daher wird der Streamingpfad komplett in Hardware gehalten. Die Bilddaten laufen zuerst in ein Sensor-Interface-Modul, in dem grundsätzliche Bildvorverarbeitungsaufgaben wie zum Beispiel Dynamikanpassung, Triggersteuerung oder Farbberechnungen anzusiedeln sind. Weitergehende Bildverarbeitung lässt sich dann optional nachschalten. Das zu übertragende Bild wird anschließend in einen Memory-Interface-Block übertragen, der die Daten in einem externen Speicher puffert, um sie bei Bedarf erneut schicken zu können (Packet Resend). Anschließend erstellt der GigE Packet Composer die GigE-Vision-Ethernet-Pakete, bevor diese über einen Ethernet MAC zum



**Bild 1:** Die Architektur einer GigE-Vision-Kamera bei Nutzung des GigE Cores von Sensor to Image.



**Bild 2:** FPGA-Implementierung von CoaXpress: Das Blockschaltbild einer CXP-Kamera ähnelt dem der GigE-Vision-Kamera.

PHY oder SFP+-Modul übertragen werden. 1G-Designs arbeiten typischerweise mit (R)GMII PHYs, wohingegen 10G-Systeme mit SFP+-Modulen arbeiten, die Transceiver benötigen.

### CoaXpress

Bei CoaXpress (*Bild 2*) sind bis zu vier Verbindungen à 1,25 bis 6,25 Gbit/s möglich; hierzu sind immer serielle Transceiver erforderlich. Da kein Pa-

(Bilder: Sensor to Image)

	USB3 Vision	CoaXpress	CLinkHS	GigE Vision
<b>Connection Type</b>	- Master-Slave - Shared Bus - 4 pin STP	- Point to Point - 75 Ohm RG59	- Point to Point - CX4 - SFP+	- Point to Point or LAN - RJ45/Cat6/7 - SFP+
<b>Data Rate</b>	<5 Gbit/s	<25 Gbit/s	<10 Gbit/s per link	<10 Gbit/s per link
<b>Transmission Distance</b>	<3 m	<200 m	<15 m (copper) <300 m (fiber)	<100 m (copper) <300 m (fiber)
<b>Scalability</b>	127 Devices	4 Devices/FG	1 Device/FG	Unlimited
<b>PC Interface</b>	Motherboard or PCIe card	PCIe Framegrabber	PCIe Framegrabber	GigE NIC

cket-Resend-Mechanismus vorgesehen ist, entfällt das Memory Interface, was die Ressourcen-Anforderungen an das FPGA reduziert. Die Transceiver werden an einen Equalizer-Baustein angeschlossen, der dafür sorgt, dass sowohl Streaming-Daten als auch Control-Da-

ten über ein Coax-Kabel übertragen werden können.

Sowohl der GigE Vision Core als auch der CoaXPress Core sind bidirektional implementiert, das heißt, sie eignen sich sowohl für Kamera- als auch für Framegrabber-Applikationen.

## Ein Kostenvergleich

Bei allen technischen Möglichkeiten spielen die Kosten eine wesentliche Rolle bei der Entscheidung für oder gegen einen Interface-Standard. Nachfolgend ein erster Überblick darüber, welche Kosten bei der Produktion einer High-End-Kamera GigE Vision 10G oder 20G gegenüber einer Kamera mit CoaXPress

mit gleicher Datenrate zu erwarten sind. Betrachtet werden die Interface-spezifischen Kosten; Kosten, die bei beiden Varianten in vergleichbarer Größe entstehen, sind nicht einbezogen. Standardspezifische Einmalkosten (NRE) werden auf die zu produzierenden Kameras umgelegt.

### Kosten für eine 10/20G-GigE-Vision-Kamera

	GigE Vision 10G SFP+	GigE Vision 20G SFP+	GigE Vision 10G CAT7	GigE Vision 20G CAT7
NRE (Core)	40.000	50.000	40.000	50.000
Anzahl Kameras in Produktion	200	200	200	200
FPGA	150	180	150	180
PHY/Connector/ SFP+ Modul	120	240	100	200
<b>Gesamt für eine Kamera</b>	<b>470</b>	<b>670</b>	<b>450</b>	<b>630</b>
Kabel 20m	30	60	25	50
Netzwerkkarte	475	950	450	900
<b>Gesamt für ein Kamerasystem</b>	<b>975</b>	<b>1680</b>	<b>925</b>	<b>1580</b>

Die Kosten für eine Kamera oder ein Gesamtsystem unterscheiden sich nicht wesentlich. Empfehlenswert ist jedoch die Glasfaser-Version (mittels SFP+-Modul), da sie deutlich weniger Verlustleistung aufweist (Interfacekosten in Euro).

### Kosten einer CoaXPress-Kamera

	2×CXP6	4×CXP6
NRE (Core)	30.000	30.000
Anzahl Kamera in Produktion	200	200
FPGA	120	120
PHY/Connector	60	120
<b>Gesamt für eine Kamera</b>	<b>330</b>	<b>390</b>
Kabel 20m	110	220
Framegrabber	800	900
<b>Gesamt für ein Kamerasystem</b>	<b>1240</b>	<b>1510</b>

Bei einer Gegenüberstellung der Kosten für eine CoaXPress-Kamera mit einer Datenrate von 10 Gigabit (2×CXP6), fällt auf, dass die kameraspezifischen Ausgaben unter denen für eine GigE-Vision-Kamera liegen. Werden jedoch die PC-seitigen Komponenten mitberücksichtigt, machen die erhöhten Ausgaben für Kabel und vor allem den Framegrabber diesen Vorteil wieder zunichte. Erst bei einem 20G-System hat CXP im Vergleich zu Ethernet im Gesamtsystem wieder die Nase vorne (Interfacekosten in Euro).

## Die USB3-Vision-Implementierung

Auch USB ist ein serielles Protokoll, bei dem üblicherweise USB-Chips wie der Cypress FX3 eingesetzt werden; hier sind also FPGA-Transceiver nicht zwingend notwendig. Bei einem Einsatz des FX3 ist denkbar, völlig ohne FPGA auszukommen, da dieser Chip auch eine ARM-CPU enthält, die die Paketgenerierung übernehmen kann. Allerdings lassen sich damit nur Datenraten von einigen 10 MByte/s erreichen, was weit von den eigentlichen Möglichkeiten von USB3 entfernt ist. Um alle Möglichkeiten auszureizen, sollte ein FPGA vorgeschaltet werden, das die Streaming-Pakete für das USB3-Vision-Protokoll generiert und zum direkten Versand an den FX3 übergibt. Der Control Channel wird wie bei der GigE-Vision- oder CoaXPress-Implementierung von der CPU bearbeitet. In der High-Performance-Variante ist der FX3 ebenfalls durch einen FPGA-Core ersetzbar, was die Leistungsfähigkeit weiter steigert.

## Eine Frage der Datenmengen

Aufgrund der immer höher werdenden Auflösungen und Bildwiederholraten moderner Sensoren werden entsprechende Schnittstellen benötigt, um die resultierenden Datenmengen übertragen zu können. Sollen zudem Protokolle wie GigE Vision, CoaXpress oder USB3 Vision implementiert werden, ist das mit CPU/Software-basierten Systemen kaum mehr realisierbar. Da FPGAs die nötigen Verarbeitungsschritte parallel durchführen können und die nötigen Hardware-Interfaces aufweisen, bieten sie sich für diese Aufgabe an. Ein weiterer Pluspunkt: Mit Hilfe entsprechender Cores und Referenzdesigns lässt sich die Entwicklungszeit deutlich reduzieren. (Ein Whitepaper zur gesamten Thematik findet sich unter <http://www.xilinx.com/applications/industrial/machine-vision.html>). *ik*



**Matthias Schaffland**

ist Entwicklungsleiter bei Sensor to Image in Schongau.