

MÖGLICHKEITEN UND VORTEILE VON FPGAS IN MOTORSTEUERUNGEN – TEIL 1/2

FPGAs beleben die Antriebstechnik

Die Anforderungen an Drive Control-Systeme steigen stetig. Hohe Regelraten und immer komplexere Bewegungsberechnungen machen es zunehmend schwierig, die Systeme mit Mikro- und Signalprozessoren zu realisieren. FPGA-basierte Lösungen erfüllen höchste Anforderungen an die Dynamik der Bewegungssteuerung. Der 1. Teil dieser Artikelserie zeigt die Möglichkeiten und Vorteile FPGA-basierter Motion Control-Systeme auf. Der 2. Teil (erscheint in der nächsten Megalink) befasst sich vorwiegend damit, wie der Umstieg auf die FPGA-Technologie problemlos gelingt.

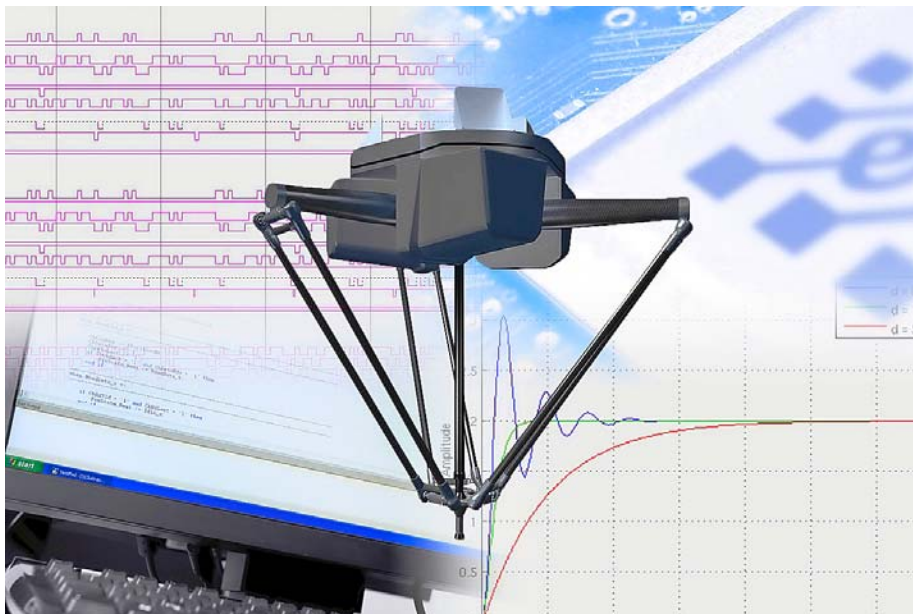


Bild 1: FPGAs eröffnen für Motion-Anwendungen neue Möglichkeiten.

(Bilder: Enclustra)

Während Motorensteuerungen bisher meist mit DSPs oder Mikrocontrollern realisiert wurden, stieg das Interesse an FPGA-Lösungen in den letzten Jahren markant an. Gründe dafür sind neben den sinkenden Kosten und der stetig steigenden Rechenleistung von Field Programmable

Gate Arrays (FPGAs) auch, dass in vielen Systemen programmierbare Logik schon für andere Zwecke wie zum Beispiel für Feldbus-Interfaces eingesetzt wird. Zudem ist eine hohe Integrationsdichte heute in vielen Systemen ein Muss. Weitere Punkte sind die Langzeitverfügbarkeit und die Möglichkeiten zur zukünftigen Weiterentwicklung, ohne dass eine Überarbeitung der Hardware nötig wird.

- Relativ komplexe Algorithmen, die allerdings nicht in Echtzeit berechnet werden müssen. Dazu gehören unter anderem die Bahnplanung, 3D-Interpolation sowie applikationsspezifische High-Level-Control- und Kommunikationsaufgaben (zum Beispiel Feldbus-Zugriffe).

Natürlich gibt es auch Fälle, in denen die Bereiche nicht so klar getrennt werden können. Dies ist zum Beispiel der Fall, wenn während der Bewegung auf ein Ereignis reagiert und innerhalb einer bestimmten Zeit eine neue Bahnplanung durchgeführt werden muss.

Für alle Echtzeitberechnungen ist es wichtig, dass diese immer genau gleich ablaufen und genau gleich lang dauern, da Jitter auf dem Regelzeitpunkt und/oder der Verarbeitungszeit die Regelgüte beeinträchtigt. Zudem ist die Rechenleistung für diese Berechnungen aus zwei Gründen kritisch: Einerseits können mit mehr Rechenleistung höhere Regelraten erzielt werden, andererseits kann die Berechnungszeit verkürzt werden, was zu einer kleineren Verzögerung im Regelkreis und dadurch mehr Phasenreserve führt. Folglich führt der Weg zu besseren Resultaten über mehr Rechenleistung. Da die anzuwendenden Algorithmen inzwischen weitgehend standardisiert sind und nicht oft angepasst werden, spielt die Technologie mit der diese umgesetzt werden, keine grosse Rolle.

Enclustra – we speak FPGA

Das FPGA Design Center Enclustra wurde 2004 gegründet. Als Mitglied bei den Partnerprogrammen aller grossen FPGA-Hersteller ist Enclustra immer auf dem neuesten Stand und verfügt über Fachwissen in verschiedenen Anwendungsgebieten wie zum Beispiel Software Defined Radio, Drive Control, Digitale Signalverarbeitung und Datenerfassung. Neben Entwicklungsdienstleistungen in den Bereichen Hardware, FPGA-Firmware und -Software bietet Enclustra auch verschiedene Produkte wie zum Beispiel FPGA-Module und IP-Cores an. Damit deckt das Portfolio von Enclustra alle mit der FPGA-Technologie verbundenen Themenbereiche ab.

Anforderungen an Motion-Control-Systeme. Die verschiedenen in Motion-Control-Systemen anfallenden Berechnungen lassen sich grundsätzlich in zwei Bereiche einteilen:

- Mehr oder weniger einfache, immer gleich bleibende Berechnungen in Echtzeit, die hauptsächlich Additionen, Multiplikationen und Begrenzungen benötigen. Beispiele hierfür sind PID-Regler, feldorientierte Kommutierung für bürstenlose Motoren oder die Rückintegration von Trajektorien.

AUTOR

Oliver Bründler
Projektmanager Enclustra GmbH

INFOS

Enclustra GmbH
8005 Zürich
Tel. 043 343 39 43
info12@enclustra.com
www.enclustra.com

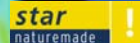
Ihre Frage: Was bietet das R&S®RTO bei 4 GHz Bandbreite?



Unsere Antwort: Die höchste Präzision und Erfassungsrate seiner Klasse.

Für die Entwicklung von digitalen, analogen und HF-Designs ist das neue R&S®RTO-Modell eine leistungsstarke Lösung: Die sehr rauscharme Eingangsstufe verfügt selbst in der kleinsten Skalierung (1 mV/Div) noch über die volle Messbandbreite von 4 GHz. Die Messdynamik ist extrem hoch (ENOB > 7 Bits), ebenso die Erfassungsrate mit 1 Million Messkurven pro Sekunde. Und die schnelle FFT-Analyse, hohe Messdynamik und eine Bandbreite bis 4 GHz ermöglichen auch Messungen im Frequenzbereich.

Mehr Informationen zu den Oszilloskopen unter:
www.scope-of-the-art.de/ad/faq/rto4



ROHDE & SCHWARZ
ROSCI ROHDE & SCHWARZ AG

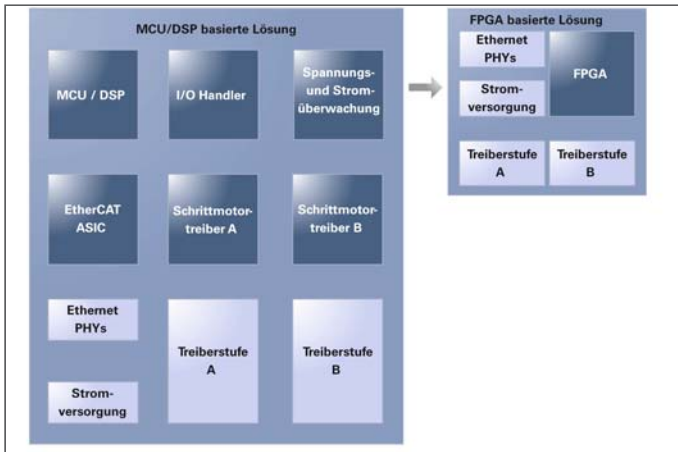


Bild 2: Schematische Gegenüberstellung einer MCU/DSP-basierten und einer FPGA-basierten Hardware.

Bei den Algorithmen, die nicht in Echtzeit ablaufen müssen, ist ein Prozessor meist unabdingbar. Operationen wie das Wurzelziehen oder das numerische Lösen von Gleichungssystemen, die in vielen Algorithmen eine wichtige Rolle spielen, lassen sich anders kaum sinnvoll realisieren. Ausserdem ist ein in C/C++ programmierbarer Prozessor für Spezialisten mit applikationsspezifischem Know-how einfach zugänglich, und genau diese Applikationsspezialisten stellen schliesslich die Qualität des Endproduktes sicher. Es ist also wichtig, dass die Implementation von Spezialisten ohne technologiespezifisches Know-how vorgenommen werden kann.

Das Beste aus zwei Welten. FPGAs besitzen viele Vorteile für Motion-Control-Systeme. Aus dem obigen Abschnitt lässt sich schlussfolgern, dass reine FPGA-Logik allein nicht alle Probleme lösen kann. Deshalb ist es in Motion-Control-Systemen üblich, dass ein oder zwei Soft-Prozessoren (siehe Kasten) ins FPGA integriert werden. Einerseits können mit ihnen alle Berechnungen ohne harte Echtzeitanforderungen durchgeführt werden, andererseits wird es dadurch möglich, das Know-how von Spezialisten zu nutzen, welche die Programmierung in C/C++ beherrschen, aber keine FPGA-Cracks sind. Für alle Echtzeitberechnungen, die im Grossen und Ganzen dem aktuellen Stand der Technik entsprechen, bringt die Umsetzung in FPGA-Logik viele Vorteile:

Rechenleistung: kurze Berechnungszeiten (wenig Verzögerung, mehr Phasenreserve), hohe Regelraten.

CPU-Unabhängigkeit: dramatisch reduzierte Interruptlast, mehr Rechenzeit für typische CPU Tasks wie zum Beispiel High-Level-Control, Bahnplanung und Kommunikation.

Konstante Rechenzeit: In FPGA-Logik lässt sich ein jitterfreies System viel leichter realisieren als mit einer CPU.

Ressourcentrennung: Mehrere Achsen können komplett unabhängig voneinander realisiert werden und beeinträchtigen einander somit in keiner Weise.

Erweiterbarkeit: Da zusätzliche Berechnungen nicht dieselben physikalischen Ressourcen benötigen wie die bestehenden Algorithmen, werden Ressourcenkonflikte verhindert und Weiterentwicklungen beeinträchtigen das bisherige System nicht.

Integrationsdichte: Dadurch, dass praktisch die ganze Funktionalität inklusive Prozessor auf einem Chip realisiert werden kann, wird nicht nur Platz auf dem PCB gespart, sondern auch die Beschaffungs- und Lagerkosten, die Abkündigungsgefahr sowie der Hardware-Entwicklungsaufwand sinken (Bild 2).

Abkündigungssicherheit: Einerseits sind FPGAs relativ lang verfügbar (im Normalfall mehr als 10 Jahre), andererseits lässt sich die gleiche Funktionalität einfach auf andere FPGAs portieren. Die Wiederverwendbarkeit der Entwicklungen ist damit gewährleistet. →

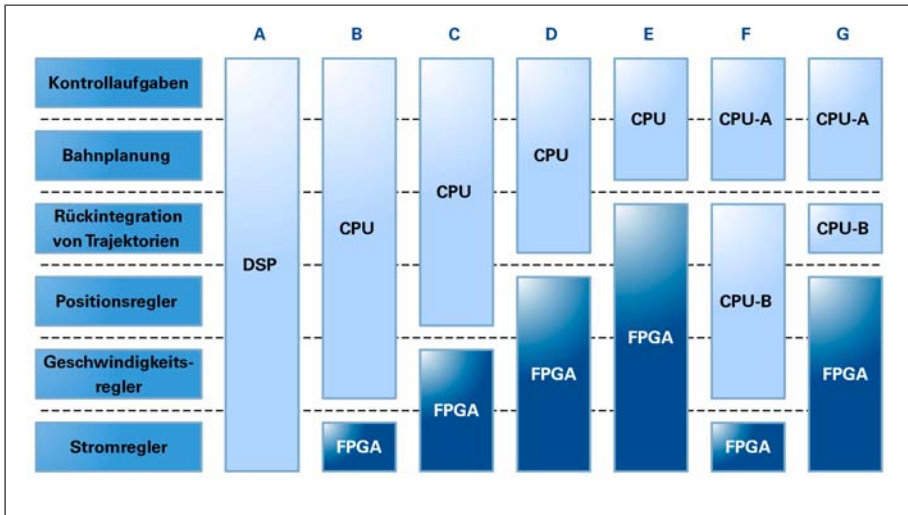


Bild 3: Mögliche Aufteilungen der in einem Motion Control-System anfallenden Aufgaben.

Der einzige nennenswerte Nachteil der FPGA-Technologie ist, dass viele bestehende Motion-Control-Spezialisten (noch) keine Erfahrung mit der Entwicklung von FPGA-Designs haben.

Flexibilität ist Trumpf. Bild 3 illustriert diverse mögliche Aufteilungen der anfallenden Aufgaben. Die Zusammenstellung ist keinesfalls abschliessend, zeigt aber die gebräuchlichsten Varianten auf.

Die reine DSP-Lösung (A) ist die bisher am weitesten verbreitete Variante. Ein DSP (oder allenfalls auch ein Mikrocontroller) übernimmt alle Aufgaben. Das Problem bei dieser Variante ist, dass für jede Reglerberechnung ein Interrupt bearbeitet werden muss, was zu vielen Tausend Interrupts pro Sekunde führt. Andere Nachteile sind, dass ein Prozessor die gesamte Rechenlast tragen muss und dass eine Erweiterung auf mehrere Achsen oft kaum möglich ist, da die Rechenleistung dafür nicht mehr ausreicht. Nicht zu vernachlässigen ist zudem, dass heute immer mehr bürstenlose Gleichstrommotoren (auch als BLDC- oder brushless Motoren bezeichnet) eingesetzt werden, deren Regelung mehr Rechenleistung braucht als etwa die Regelung von DC-Motoren.

Die Varianten B bis D zeigen die Auslagerung der Regelkreise in FPGA-Logik. Alle anderen Aufgaben werden weiterhin von der CPU erledigt. Wenn schon Regler im FPGA realisiert worden sind, ist der Aufwand, um weitere Regler in FPGA-Logik zu realisieren, relativ klein. Dies lässt den Schluss zu, dass Variante B (nur Stromregler in FPGA-Logik) gegenüber Variante D (Positions-, Geschwindigkeits- und Stromregler in FPGA-Logik) wenig Sinn macht. Die Varianten B (nur Stromregler in FPGA-Logik) und C (Geschwindigkeits- und Stromregler in FPGA-

Logik) haben allerdings den Vorteil, dass für die Positions- und/oder Geschwindigkeitsregelung auch komplexere Algorithmen in C/C++ implementiert und von Nicht-FPGA-Spezialisten bearbeitet werden können. Generell lässt sich sagen, dass bereits die Auslagerung des Stromreglers in FPGA-Logik zu einer grossen Entlastung führt, da dieser die höchste Regelrate aufweist und damit tendenziell zur grössten Interrupt- und Rechenlast führt. Zusätzlich beinhaltet der Strom-Regelkreis für Brushless-Motoren auch die gesamte feldorientierte Kommutierung, die sehr rechenintensiv ist (Bild 4).

Im Fall E (Bild 3) wird auch die Rückintegration von Trajektorien in FPGA-Logik ausgeführt. Dies bringt auf der einen Seite den Vorteil mit sich, dass keine Realtime-Anforderungen an die CPU bestehen. Auf der anderen Seite ist die Implementation im FPGA relativ komplex, da oft Floating-Point-Berechnungen notwendig sind.

Die Varianten F und G unterscheiden sich von B und D nur dadurch, dass je ein einzelner Prozessor für Realtime (CPU-B) und Non-Realtime (CPU-A)-Aufgaben verwendet wird. Dadurch bestehen an den Prozessor CPU-A keine Realtime-Anforderungen und trotzdem sind im Realtime-Bereich komplexere Algorithmen in weitverbreiteten Programmiersprachen wie C/C++ realisierbar.

Was die Make-or-Buy-Entscheidung betrifft, lässt sich sagen, dass fertige Lösungen (sogenannte IP-Cores) für die Implementation von Reglern für verschiedene Motortypen inklusive feldorientierter Regelung verfügbar sind (unter anderem der Universal Drive Controller IP-Core der Firma Enclustra GmbH). Zusammen mit der Tatsache, dass Soft-Prozessorsysteme mit den Tools der FPGA-Hersteller per Mausclick sehr einfach zusammengestellt werden können, lassen sich die Varianten B/C/D/F/G (Realisierung der Regelkreise in FPGA-Logik) ohne viel FPGA-Kenntnisse umsetzen und nutzen. Der tatsächliche Implementationsaufwand kann dadurch sogar deutlich geringer sein, als für die reine DSP-Variante A. Sobald

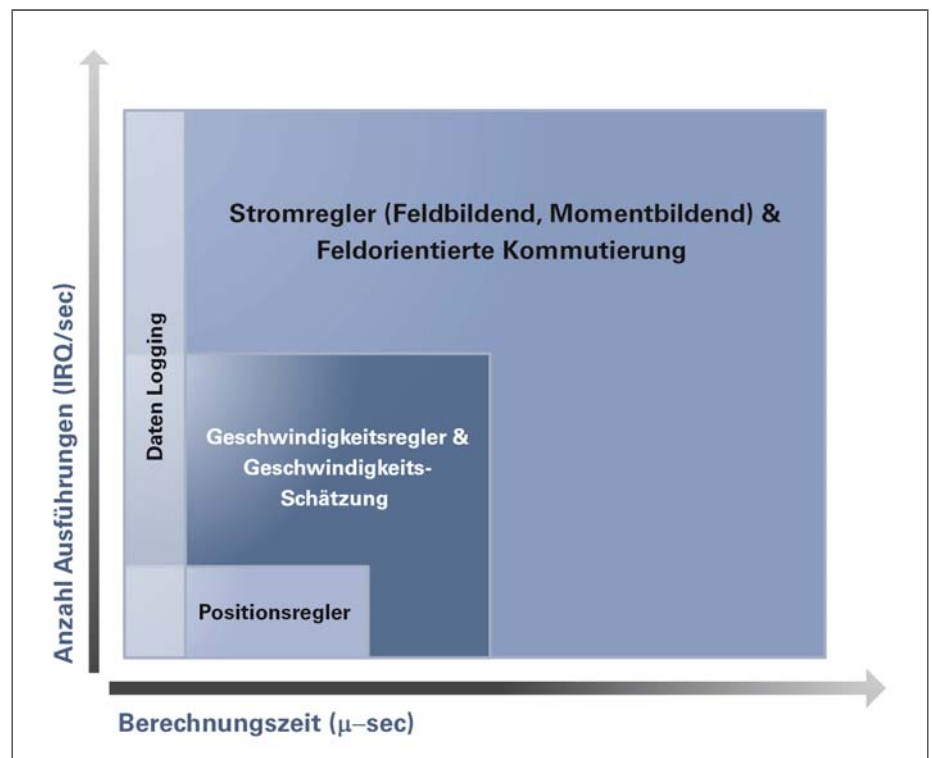


Bild 4: Benötigte Rechenzeit für die verschiedenen Realtime Tasks einer BLDC-Regelung.

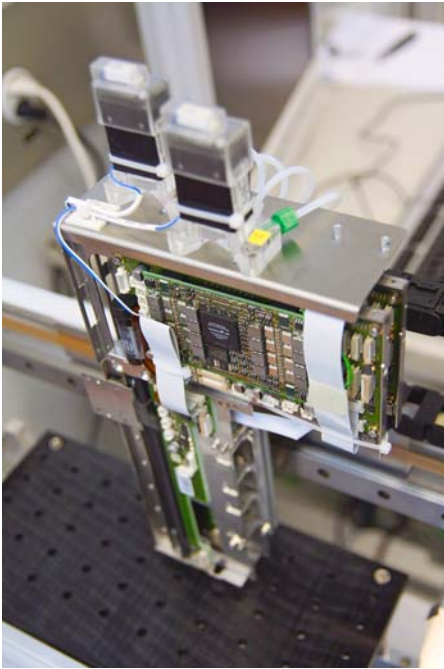


Bild 5: Das beschriebene Motion Control-Modul wird in Robotik-Anwendungen eingesetzt.

aber komplexere Algorithmen in FPGA-Logik realisiert werden, wie dies in Variante E (Trajektorien Rückintegration in FPGA-Logik) angedeutet ist, wird in den meisten Fällen eine applikationsspezifische Entwicklung von FPGA-Logik nötig.

Die Theorie in die Praxis umgesetzt. Dass ein FPGA-basiertes Motion Control-System leistungsfähig und auch finanziell sinnvoll realisierbar ist, zeigt ein vom FPGA-Design-Center Enclustra realisiertes Kundenprojekt aus der Medizinaltechnik.

Die Anforderungen an das beschriebene System sind hoch: Es werden bis zu vier Achsen gleichzeitig geregelt, wobei BLDC-, DC- und Schritt-Motoren verwendet werden

und Reglertakte bis zu 100 kHz möglich sind. Da in einem Gerät Dutzende dieser Einheiten an einen Feldbus (in diesem Fall CANopen) angebunden werden, muss die Kommunikation auf ein Minimum beschränkt werden, um dessen Kapazitätsgrenze nicht zu überschreiten.

Die Lösung für dieses Problem liegt auf der Hand: Die gesamte Motion Control-Intelligenz wird nicht in einem zentralen Rechner, sondern direkt in den Endgeräten realisiert. Das als Herzstück des Systems verwendete Altera Cyclone III FPGA enthält deshalb zwei 80 MHz schnelle Nios II 32-Bit-Soft-Prozessoren. Der erste Prozessor ist für alle Aufgaben zuständig, die keine harten Realtime-Anforderungen aufweisen. Darunter fallen unter anderem die Feldbus-Kommunikation und die Berechnung von Bahnprofilen. Der zweite Prozessor übernimmt die Ausführung der Fahraufträge, also die Verfolgung eines Profils in Echtzeit, sowie die Positions- und Geschwindigkeitsregelung, beides für alle vier Achsen. Beide Prozessoren kommunizieren über einen gemeinsamen Speicher (shared Memory). Beide Prozessoren wurden mit einer Fließkomma-Recheneinheit ausgestattet, sodass alle Berechnungen ohne Umrechnung direkt in SI-Einheiten durchgeführt werden können. Die spezifischen Vorteile der FPGA-Technologie – massiv parallele Datenverarbeitung, viel Rechenleistung und hohe zeitliche Auflösung – ermöglichen es, die geforderte Dynamik der Regelkreise zu erreichen. Direkt in FPGA-Logik ausgeführt werden die mit 100 kHz sehr schnellen Stromregler, die ebenso schnelle feldorientierte Regelung der BLDC-Motoren sowie einige zusätzliche, applikationsspezifische Wünsche des Kunden. Dazu gehören unter anderem eine Flüssigkeitsniveauserkennung, eine Schrittverlustüberwachung und die kontinuierliche Abschätzung der aktuellen Motortemperatur.

Soft-CPU in FPGAs

Von einer Soft-CPU spricht man, wenn ein Teil der im FPGA vorhandenen Logik-Ressourcen dazu verwendet wird, einen Prozessor zu realisieren. Während solche Soft-CPU von den Taktraten her (bis rund 100 MHz) nicht mit herkömmlichen, fest in Silizium realisierten Prozessoren (mehrere 100 MHz) mithalten können, sind sie bezüglich Architektur und Features durchaus vergleichbar. So sind beispielsweise FPUs, MMUs und MPUs verfügbar und die meisten heute eingesetzten Soft-CPU sind 32-Bit-RISC-Prozessoren. Dies ermöglicht es, herkömmliche Betriebssysteme wie Linux problemlos auszuführen. Der grosse Vorteil von Soft-CPU liegt darin, dass Peripherien, Features oder der Instruktionssatz auf die Applikation hin optimiert werden können und dass ein externer Prozessor eingespart werden kann. Wichtig ist auch, dass die Kommunikation zwischen Prozessor und FPGA-Logik auf dem Chip stattfindet und daher der traditionelle Kommunikations-Flaschenhals CPU-FPGA überwunden ist.

Interessant ist auch das Modell der Zusammenarbeit mit dem Kunden: Enclustra entwickelte die gesamte FPGA-Logik und programmierte den Prozessor, der für die Echtzeit-Aufgaben zuständig ist. Der zweite Prozessor, auf dem die Software läuft, die das gesamte applikationsspezifische Know-how des Kunden enthält, wurde vom Kunden programmiert und verhält sich für ihn gleich, wie ein Prozessor aus dem Katalog – allerdings mit hoch optimierten und perfekt auf die Anforderungen zugeschnittenen Peripherien. Durch diese Arbeitsteilung kann der Kunde das Kern-Know-how im Haus behalten und sich auf die aus seiner Sicht wirklich wichtigen Dinge konzentrieren. Lesen Sie in der nächsten Megalink den zweiten Teil dieses Artikels der erläutert, wie der Einstieg in die FPGA-Technologie reibungslos gelingt. ■

Schlauchpakete ab 24 h
Leitungen blitzschnell austauschen. Ausfallzeiten senken.

... für Roboter
Einzel oder komplett konfektioniert.

Torsionsleitungen ab Lager.

...igus.ch/triflexR
Tel. 062 388 97 97 Fax 062 388 97 99 Mo.-Fr. 8.00 bis 20.00h Sa. bis 12h Besuchen Sie uns: Intersolar - Halle B6 Stand 151