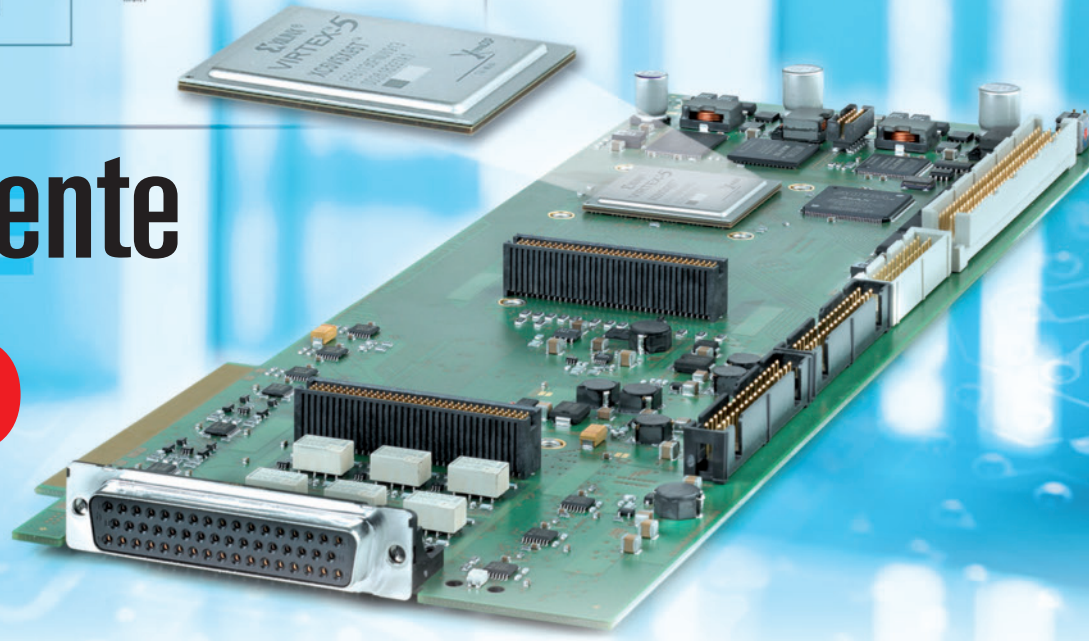


EINSATZ VON FPGAS FÜR RAPID PROTOTYPING UND HiL-TESTS

Intelligente I/O



Stetig steigende Anforderungen an die Rechenleistung von Rapid Control Prototyping- und HiL-Systemen erfordern es, dass Funktionen, die früher als Software auf Prozessoren berechnet wurden, heute in Hardware bearbeitet werden müssen. FPGAs bieten die Möglichkeit, dies mit nahezu der Flexibilität von Prozessoren und dem Geschwindigkeitsvorteil parallel arbeitender Hardware durchzuführen.

Neue Abgasnormen und Gesetze zur Verbrauchsverringern bei Kraftfahrzeugen verlangen nach neuen Antriebskonzepten und diese wiederum häufig nach einer anspruchsvolleren Sensorik. Nicht zuletzt sind davon auch die Entwicklungssysteme für Simulation und Test der neuen Regelsysteme betroffen – das Rapid Control Prototyping (RCP) und die Hardware-in-the-Loop (HiL)-Simulation. Für die Unterstützung der neuen Sensor-Schnittstellen und für die erhöhten Simulationsanforderungen bietet der Einsatz durch den Anwender programmierbarer FPGAs direkt an der I/O-Schnittstelle die Möglichkeit, erfasste Daten vorzuverarbeiten, aber auch sehr schnelle Regelschleifen direkt zu schließen, um somit zum Beispiel der hohen Dynamik von Elektromotoren gerecht zu werden.

Field Programmable Gate Arrays (FPGAs) sind programmierbare logische Schaltungen. Bei der Programmierung werden im Wesentlichen Logikelemente vielfältig miteinander verschaltet. Ein Logikelement besteht dabei aus einer programmierbaren Wahrheitstabelle (Look-up Table =

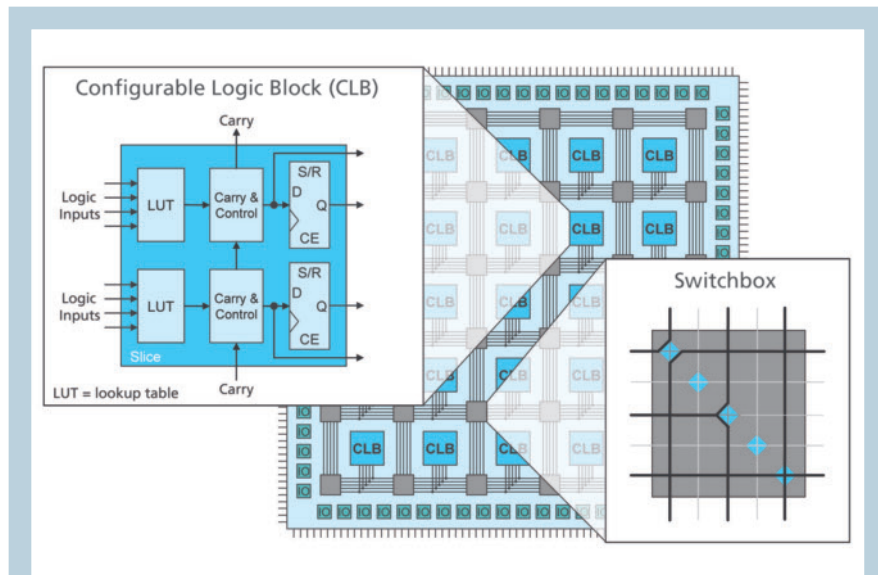
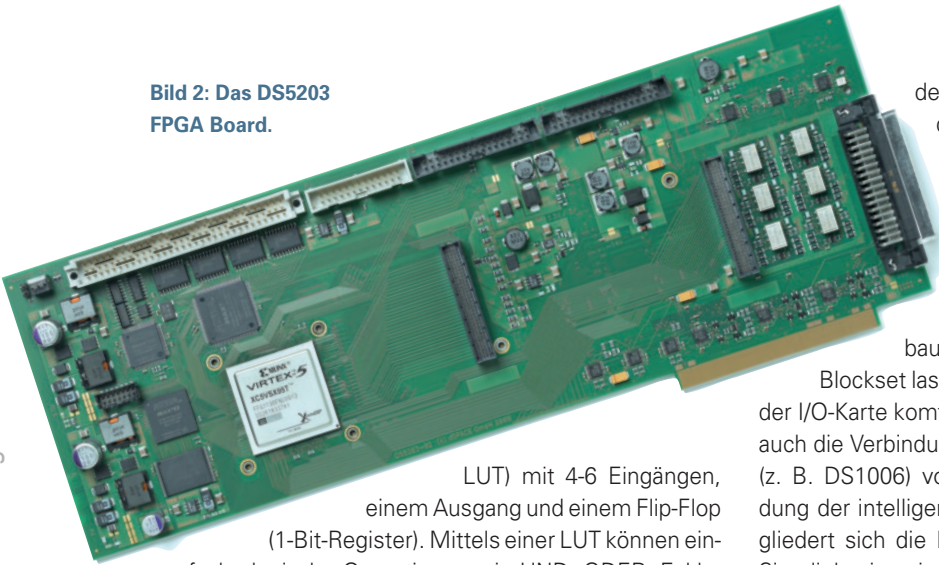


Bild 1: Schematischer Aufbau eines FPGAs.

Bild 2: Das DS5203 FPGA Board.



LUT) mit 4-6 Eingängen, einem Ausgang und einem Flip-Flop (1-Bit-Register). Mittels einer LUT können einfache logische Operationen wie UND, ODER, Exklusives ODER usw. realisiert werden. Durch die Zusammenschaltung mehrerer LUTs lassen sich beliebig komplexe logische Zusammenhänge abbilden, wobei alle Logikelemente parallel arbeiten (Bild 1). Um dem Anwender die Konfiguration der LUTs und deren Verschaltung zu ermöglichen, gibt es Hardware-Beschreibungssprachen wie VHDL. Diese ermöglichen es, logische Zusammenhänge oder Verhaltensmuster textuell zu beschreiben, ohne dass der Aufbau des FPGAs bekannt sein muss. Mit einem Synthesewerkzeug des FPGA-Herstellers kann dann aus der VHDL-Beschreibung die Konfiguration des FPGAs erzeugt werden.

Aus dem Simulink-Modell direkt auf das FPGA

Um die Programmierung weiter zu vereinfachen, gibt es seit einiger Zeit Werkzeuge für die grafische Modellierung des Verhaltens eines FPGAs, zum Beispiel den Xilinx System Generator (XSG), ein Simulink-Blockset zur Konfiguration von Xilinx-FPGAs. Der XSG bietet dabei neben einfachen logischen Elementen auch komplexe Blöcke an, wie Fourier-Transformationen oder FIR-Filter, die eine Vielzahl von Logikelementen nutzen. Um das XSG-Modell mit

den Schnittstellen des FPGAs zu verbinden, bietet dSPACE sowohl die benötigte Hardware (DS5203 FPGA Board (Bild 2) als auch zugehörige Software (RTI FPGA Programming Blockset) an. Das DS5203 FPGA Board enthält einen durch den Anwender programmierbaren Xilinx Virtex-5 FPGA und I/O-Treiberbausteine. Mit dem RTI FPGA Programming Blockset lassen sich sowohl die I/O-Treiberbausteine der I/O-Karte komfortabel an das XSG-Modell anbinden als auch die Verbindung des Modells zu einer Prozessorkarte (z. B. DS1006) von dSPACE modellieren (siehe „Einbindung der intelligenten I/O in das Echtzeitsystem“). Somit gliedert sich die Programmierung des FPGAs nahtlos in Simulink ein, ein Vorgang, der dem dSPACE-Anwender bereits für die Konfiguration von Prototyping-Hardware im Rahmen der modellbasierten Reglerentwicklung oder der HiL-Simulator-Hardware und deren Verbindung mit den Streckenmodellen für den Steuergerätestest bekannt ist. Auch beim Einsatz des FPGA-Boards können Synthese, Build-Prozess und Programmierung des FPGAs bzw. des Prozessors direkt aus Simulink gestartet werden, sodass sich die Handhabung des FPGAs genauso komfortabel darstellt wie die bisherigen RCP- und HiL-Szenarien. Allerdings gibt es bei der Erstellung von FPGA-Applikationen einiges mehr zu beachten. Da jede Funktion und jeder Modellblock durch die Synthese auf dedizierte Hardware abgebildet wird, benötigt auch jedes Bit jedes Signals Platz auf dem FPGA. Dementsprechend ist es auch nur möglich, mit Festpunktdatentypen zu arbeiten, bei denen die Bit-Breite und die Kommastelle dem Wertebereich des Signals angepasst werden müssen. Hierbei ist wichtig, dass sich zum Beispiel bei der Multiplikation zweier Signale die Bit-Breite des resultierenden Signals als Addition der beiden Bit-Breiten der Eingangssignale ergibt. Um nicht zu breite Signale zu erhalten, muss der Signaltyp wieder an den zu erwartenden Wertebereich angepasst werden. Weiterhin kann es vorkommen, dass durch die parallele Bearbeitung

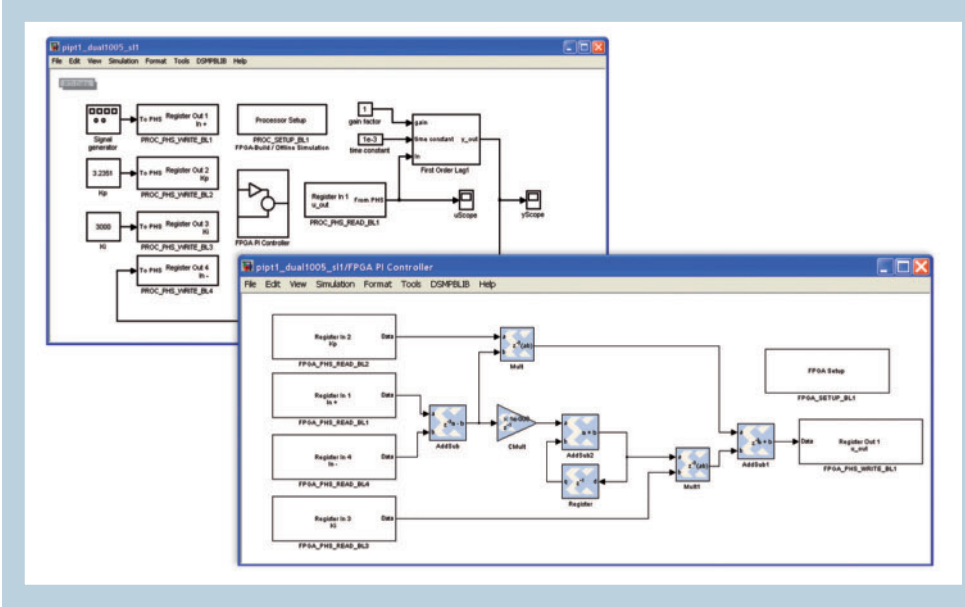


Bild 3: Einbindung des XSG-Modells am Beispiel eines PI-Reglers.

www.hanser-automotive.de Nicht zur Verwendung in Intranet- und Internet-Angeboten sowie elektronischen Verteilern. © 2009 Carl Hanser Verlag, München

aller Signalpfade die einzelnen Signale unterschiedlich lange Laufzeiten haben. Sollen solche Signale miteinander verknüpft werden, muss das schnellere Signal entsprechend verzögert werden, sodass die Signale die gleiche Laufzeit haben. Dieses Verhalten wird bereits bei der Simulink-Offline-Simulation akkurat nachgebildet, sodass die Probleme schon mit Hilfe dieser analysiert und gelöst werden können.

Einbindung der intelligenten I/O in das Echtzeitsystem

Es ist wahrscheinlich, dass Programme, die eine FPGA-Unterstützung benötigen, in einem interdisziplinären Team mit einem Software-Entwickler und einem FPGA-Experten erstellt werden. **Bild 3** zeigt ein Beispiel, in dem der FPGA-Experte einen PI-Regler auf dem FPGA implementiert hat und das erste Zwischenergebnis in der Offline-Simulation überprüft. Im nächsten Schritt wird die Implementierung auf dem FPGA mit einer Simulation der Umgebung auf der angebundenen Prozessorkarte getestet, bevor die reale Strecke mit der I/O des DS5203 verbunden wird. Diese Anbindung muss analog auch im FPGA-Modell mit Hilfe der dSPACE RTI-FPGA-Blöcke vorgenommen werden. Gleichzeitig arbeitet der Software-Entwickler an der übergeordneten Regelung, die später auf der Prozessorkarte ausgeführt wird. Zur Funktionsüberprüfung können die beiden Entwickler ihre Zwischenstände immer wieder offline simulieren.

Präzise Simulation von E-Motoren

Soll ein Steuergerät für einen Elektromotor (z. B. beim Hybridantrieb) an einem HiL-Simulator getestet werden, muss ein Modell des E-Motors mit dem Simulator abgebildet werden. Bei E-Motoren wird der Leistungsfluss direkt vom Steuergerät kontrolliert und üblicherweise mit 20 kHz angesteuert. Eine Modellimplementierung, die den hohen Dynamikanforderungen gerecht wird und die Effekte an den Leistungsendstufen modelliert, muss Zykluszeiten von deutlich unter einer Mikrosekunde erreichen. Dies kann nur mit Hilfe einer Implementierung in einem FPGA erreicht werden, indem zumindest die Motorstromberechnung als Teilmodell auf die FPGA-Karte ausgelagert wird.

DS5203 FPGA Board

Das neue DS5203 bietet einen sehr leistungsfähigen frei programmierbaren FPGA, den Xilinx Virtex-5 SX95 FPGA mit 94298 Logikzellen und 640 speziellen DSP-Blöcken. Diese DSP-Blöcke ermöglichen u. a. die Multiplikation von zwei Signalen in nur einem DSP-Block.

Weiterhin bietet die I/O-Karte 6 AD-Wandler, 6 DA-Wandler (14 bit, 10 MSample/s) und 16 digitale I/O-Kanäle sowie die Anbindung an eine Prozessorkarte per PHS-Bus. Reichen die I/O-Kanäle nicht aus, ist es möglich, ein I/O-Modul auf das DS5203 aufzustecken und somit die Anzahl der I/O-Kanäle

passend für die jeweilige Anwendung zu erweitern. Die Taktung des FPGAs mit 100 MHz ermöglicht es dem Anwender, auch anspruchsvolle Aufgaben mit kurzen Zykluszeiten auf dem FPGA zu implementieren.

Fazit

Die Programmierung des DS5203 FPGA-Boards mit dem RTI FPGA Programming Blockset und dem Xilinx System-Generator bietet dem Anwender nun die Möglichkeit, in einem Simulink-Modell eine Echtzeit-Processor-Applikation und eine FPGA-Applikation zusammen zu entwickeln. Dabei kann deren Zusammenspiel schon in der Offline-Simulation getestet werden, bevor die beiden Applikationen zusammen auf das Echtzeitsystem geladen werden. Durch diesen durchgängigen Workflow kann der Anwender flexibel und schnell auf steigende Anforderungen bei der Signalvorverarbeitung, der Nutzung neuer Interfaces oder der Beschleunigung von Modellteilen reagieren. (oe)



Dipl.-Ing. Jürgen Klahold ist Produktingenieur für Hardware-in-the-Loop-Simulatoren bei der dSPACE GmbH.

