

Digitale Logikkerne – Technik und Anwendung

Mit FPGA und Prozessor schnell zum lauffähigen Produkt

Der Trend bei der Entwicklung mikroprozessorgesteuerter Systeme geht hin zum Einsatz von Standard-Baugruppen. Damit sollen Entwicklungszeit und -kosten für die Hardware gespart werden. Aber Standard-Bauteile sind nicht für bestimmte Anwendungen optimiert. Ein frei programmiertes System, das bereits ein grundlegendes, lauffähiges Funktionsgrundgerüst mitbringt, hilft, die Entwicklungszeit bei anwendungsspezifischer Hardware zu verkürzen.

Von Peter Seng

Die Ansprüche an die Leistungsfähigkeit elektronischer Systeme haben sich in den letzten Jahren stark erhöht. Die erzielbaren Erlöse konnten mit den erheblich gestiegenen Aufwendungen für die Entwicklung dieser Systeme nicht Schritt halten. Die scheinbare Lösung dieses Problems liegt in der Massenproduktion und der Anwendung von hochintegrierten Standardsystemen, um die entstehenden Entwicklungskosten auf eine möglichst große Anzahl von Geräten zu verteilen. Bei Systemen dieser Art bestimmen letztendlich die Produktionskosten den Marktpreis.

Dies führt jedoch bei der Entwicklung für Marktsegmente mit kleinem und mittlerem Produktvolumen zu Schwierigkeiten. Die Lösung der eigentlichen Aufgabenstellung tritt gegenüber der Kostenfrage in den Hintergrund. Viele Entwickler stehen vor der Wahl, entweder ein kostengünstiges, jedoch für die Applikation meist nicht optimales Standardprodukt einzusetzen und dieses zeitaufwendig zu adaptieren, oder den Weg der aufwendigen Eigenentwicklung zu gehen. Die entstehenden Kosten sind im Voraus schlecht abzuschätzen, bzw. es ist nicht gewährleistet, ob das entstehende Produkt die Stückzahl und Lebenszeit erreicht, die eine solche Entwicklung rechtfertigt. Dies führt in vielen Fällen zu Systemlösungen, die weder für den

Kunden noch den Anbieter das angestrebte Ziel erreichen.

► Lange Wunschliste – mit Standards kaum erfüllbar

Für den Digitalteil eines Systems werden meist folgende Eigenschaften angestrebt:

- Überschaubarer Entwicklungsaufwand.
- Berechenbare Kosten.
- Einsatz möglichst weniger Spezialteile.
- Gewährleistung der Produzierbarkeit des Systems über einen möglichst langen Zeitraum.
- Wartbarkeit und Erweiterbarkeit des Systems beim Kunden über eine Standardschnittstelle zum PC.
- Produzierbarkeit unter Vermeidung kostenintensiver Entwicklungs- und Produktionswerkzeuge.
- Flexibilität bei der Auswahl von CPU, Speicher, Systemleistung usw.
- Verwendung bereits bekannter Komponenten und Software.
- Flexible Möglichkeit der Aufteilung der Systemaufgaben auf Hard- und Software.
- Skalierbarkeit.
- Möglichkeit des Datenaustausches des Systems mit einem PC.
- Direkte Zugriffsmöglichkeit eines externen PC auf die systeminterne Logik.

- Steuerung des Systems von einem PC.
- Eigenständiges Booten des Systems ohne PC.
- Kompatibilität zur Welt der PCs bei weitgehender Unabhängigkeit von PC-Betriebssystemen, -Hardware und -Schnittstellen mit ihren für die industrielle Anwendung viel zu kurzen Lebenszeiten.

Ein Teil dieser Wünsche kann durch frei programmierbare Steuerungen, PCs mit entsprechenden Erweiterungen oder durch vorgefertigte Mikrocontrollermodule erfüllt werden. Oft jedoch scheitert aber der Einsatz dieser Techniken an deren mangelnder Flexibilität, zu geringer Leistungsfähigkeit, zu hohen Kosten, zu großem Entwicklungsaufwand oder schlicht an den nicht erreichbaren Zielsetzungen.

► Mit frei programmierbarer Logik zum Optimum

Durch den Einsatz programmierbarer Logik in Verbindung mit einer ebenfalls frei programmierbaren CPU kann man die oben beschriebenen Zielsetzungen erreichen und den Wunsch nach einem flexiblen und leistungsfähigen System erfüllen. Dieser Ansatz eignet sich systembedingt auch für die bereits angesprochenen kleinen und mittleren Produktionsvolumina, in besonderem Maße sogar für die Entwicklung von Prototypen. An den Einstieg in diese Technik waren aber lange Zeit große, von ihrem Umfang her schlecht abschätzbare Anfangshürden gebunden. Sie fand deshalb bis jetzt keine breite Anwendung.

Diese Technik fordert prinzipiell die Lösung folgender Grundaufgaben:

- Realisierung einer internen Programmierschnittstelle zum PC, um die internen FPGAs, (C)PLDs und den CPU-Programmspeicher ohne spezielle Programmiergeräte programmieren und verifizieren zu können. Diese sollte kompatibel zu einer bereits vorhandenen PC-Schnittstelle sein.
- Treiber und Software, um auf diese Schnittstelle unter verschiedenen PC-Betriebssystemen zugreifen zu können.

- Systeminterne, nichtflüchtige Automaten, um ein selbstständiges Konfigurieren und Booten des Systems zu ermöglichen.
- Ein geeignetes, flexibles Systemverbindungs-Konzept, um diese Technik nicht auf bestimmte CPU-, FPGA- und (C)PLD-Typen und -Familien einzuschränken.
- Die Möglichkeit der Emulation bestehender FPGA/(C)PLD-Programmierschnittstellen, um Kompatibilität und somit Verfügbarkeit zu bereits bestehenden Produkten zu gewährleisten.

► Prozessor und programmierbare Logik

Der „digital logic kernel“ ist eine einfach anzuwendende Technologie, die es erlaubt, digitale Grundfunktionen – bestehend aus jeglicher Art von CPU mit externem Programmspeicher, programmierbarer Logik und Flash-Speicher – aufzubauen und zu programmieren. Hierzu werden keinerlei Programmiergeräte oder bereits vorprogrammierte Bauteile benötigt. Der „digital logic kernel“ kann selbstständig booten. Der Zugriff erfolgt über eine

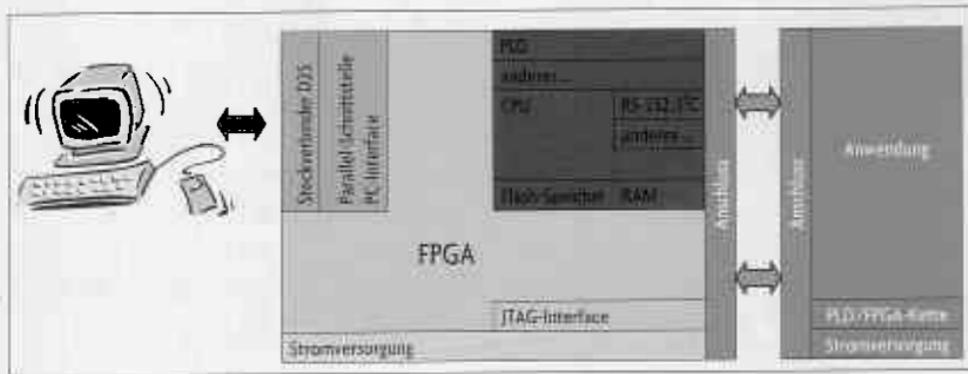


Bild 1. Der „digital logic kernel“ dlk besteht aus Prozessor, Speicher, FPGA und PC-Interface. Alles zusammen ergibt ein eigenständiges, bootbares System.

- Eine einfach erweiterbare Programmoberfläche zur komfortablen Bedienung.
- Die Verfügbarkeit dieser Lösung als offenes System, sowohl in einem sofort einsetzbaren Format als auch im vollständigen Quellcode, um eine langfristige Systempflege zu gewährleisten.

Diese Grundaufgaben können mit dem hier vorgestellten System abgedeckt werden. Es handelt sich um eine Entwicklungsumgebung bzw. Implementierung, die wir „digital logic kernel“ oder kurz „dlk“ nennen. Die Realisierbarkeit des genannten Konzeptes wird hiermit stark vereinfacht, in vielen Fällen kann sogar sofort mit der Verwirklichung der eigenen Anwendung auf einer bereits erprobten und verfügbaren Hardware begonnen werden. Eine Entwicklung unter Verwendung eines „dlk“ beschränkt sich im Wesentlichen auf den Entwurf der applikations-spezifischen Logik und des CPU-Programmcodes. Hierzu stehen jedoch erheblich preiswertere, komfortablere und leistungsfähigere Werkzeuge bereit als die heute vielfach eingesetzten.

integrierte parallele PC-Druckerschnittstelle zum Zweck der Administration und des Datenaustausches (Bild 1).

Ein Standard-PC mit einer parallelen Druckerschnittstelle ist das einzige Werkzeug, das zum Aufbau, der Programmierung und Wartung notwendig ist. Der „digital logic kernel“ wird mit einer Entwicklungsumgebung der Firma SENG digitale Systeme GmbH (www.seng.de) unter Verwendung von Halbleitern des Herstellers Xilinx und dessen Design-Software „ISE“ aufgebaut.

Die Entwicklungsumgebung ist in verschiedenen Lizenzmodellen bis hin zum vollständigen Quellcode verfügbar. Der Quellcode deckt sowohl die PC-Seite der Software unter verschiedenen Betriebssystemen wie Windows 9x, NT 4.0, 2000 (und dazu kompatibel) sowie die systemeigenen PLD- und FPGA-Ressourcen ab. Das parallele Interface zum PC und die JTAG-Emulation sind ebenfalls im Quellcode verfügbar.

Das System eignet sich für jede Art von CPU mit externem Programmspeicher, einschließlich so genannter

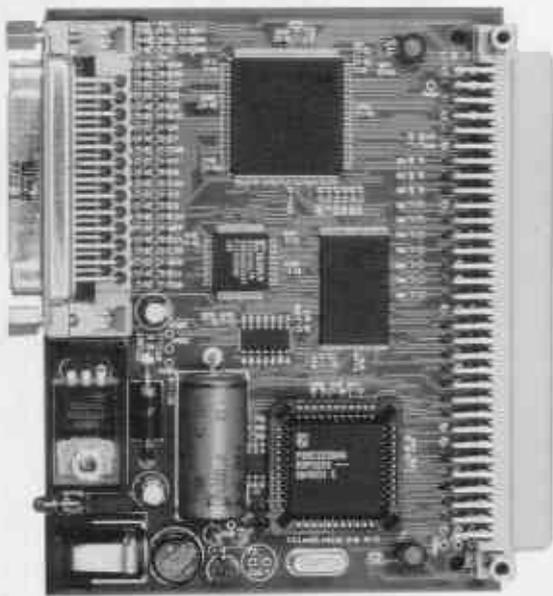


Bild 2. Eine konkrete Implementierung eines dlk-Systems besteht aus einem 8032-Controller auf einer Leiterplatte von 80 mm x 100 mm mit Parallel-Schnittstelle für die Kommunikation mit dem Entwicklungs-PC.

„Soft-CPU“ – skalierbare Prozessoren, die im FPGA-Bereich des Systems realisiert werden können. Die Bus- und Leistungsbandbreite der CPU ist in beiden Fällen frei wählbar, es kann also eine 4-bit- oder 32-bit-CPU verwendet werden – je nach Anforderung. Ein Multiprozessorbetrieb ist machbar. Bei entsprechender Auslegung kann der Logikbereich des Systems und seine interne Busstruktur sowohl von einem externen PC als auch von der internen CPU im Wechsel verwaltet werden. Diese Eigenschaft, den Busmaster wechseln zu können, bietet vor allem während der Entwicklung des Systems große Vorteile. So können z.B. auch hardwarenahe Programmteile komfortabel auf dem PC entwickelt und dann auf das Zielsystem übertragen werden. Die Steuerung des Systems gänzlich über den PC ist ebenfalls möglich. Das Entwicklungssystem beansprucht zur Laufzeit keine oder nur geringste Ressourcen im verwendeten FPGA, dieses ist also vollständig für die Applikation verfügbar. FIFO- oder dual ported Memory können auch nachträglich im FPGA realisiert werden.

Die Definition der kundenspezifischen Aufgabe kann über Schaltplan, VHDL, Verilog oder ABEL unter Verwendung der Xilinx-Entwicklungsumgebung „ISE“ erfolgen. Diese äußerst mächtige und komfortable Programmierumgebung ist kostenfrei unter der Bezeichnung „ISE WebPACK“ erhältlich. Die hiermit erzeugten Konfigurationsdateien können direkt über die parallele Schnittstelle auf das System über-

tragen oder in dessen Flash-Bereich abgelegt werden, um ein selbstständiges Booten des Systems zu ermöglichen. Dasselbe gilt für den Programmcode des verwendeten Prozessors. Die Emulation des Xilinx-JTAG-Downloadkabels (DLC5) kompatibel zur „IMPACT“-Programmiersoftware ist ebenfalls möglich. Das System kann also auch mit den Werkzeugen von Xilinx erreicht werden. Das im System befindliche CPLD ist skalierbar; d.h., falls es das System erfordert, kann es weitere Aufgaben übernehmen oder auch durch einen Baustein höherer Kapazität ersetzt werden. Besondere Vorgaben für die Verwendung bestimmter FPGA- oder (C)PLD-Serien existieren nicht, die Verwendung 5-V-toleranter Bausteine ist jedoch in vielen Applikationen sicherlich von Vorteil. Der Kern des sichtbaren Systems besteht aus einem FPGA, einem (C)PLD, einem Flash-Speicher, einer CPU (oder Soft-CPU) und einem Standard-Logik-IC der Reihe 74xxx. Je nach verwendeter CPU ist evtl. noch weiterer Speicher in Form von SRAM oder DRAM notwendig.

► Vielseitige Anwendung

Für Anwendungen erstreckt sich ein nicht übersehbares Feld von Möglichkeiten. Der Einsatz als Entwicklungsplattform, die Implementierung digitaler Kernfunktionalitäten für jede Art von Applikation, die schnelle Prototypent-

wicklung, Testanwendungen, die über eine Schnittstelle mit einem PC kommunizieren müssen, Emulationsanwendungen, die Crossentwicklung von Controller-Hardware auf dem PC und die Anwendung für Ausbildungszwecke seien hier nur stellvertretend genannt. Auf die Möglichkeit der Gestaltung so genannter Chamäleon-Systeme sei ebenfalls verwiesen. Falls also zum Zeitpunkt des Entwurfes das künf-

tig zu erfüllende Lastenheft noch nicht vollständig bekannt ist oder die Applikation oder Normung Veränderungen unterworfen sein wird, bietet das „dlk“-Konzept ebenfalls die geforderte Flexibilität und Wandelbarkeit.

Verfügbar ist ein „dlk51“-System mit Spartan2-FPGA, 8032-CPU und 512 Kbyte Flash-Speicher auf einer 80 mm x 100 mm großen Leiterplatte, versehen mit Standard-Steckverbindern und integrierter Stromversorgung (Bild 2). Das FPGA besitzt bestückungsabhängig eine Komplexität von 50 000 oder 100 000 „System Gates“. Der 8032 wurde für die erste Implementierung des „dlk“ ausgewählt, weil er einer der populärsten Prozessoren ist und eine große Vielzahl von Entwicklungswerkzeugen und Programmquellen frei verfügbar ist. Das Board eignet sich auch für die Interpretation von Soft-CPU. C-Compiler (SDCC, unter GPL Lizenz) und C-Programmbeispiele werden mitgeliefert. Das System ist al-

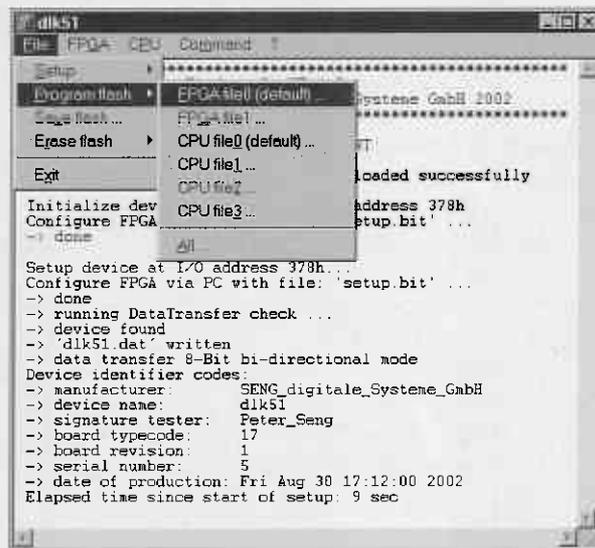


Bild 3. Die Windows-Entwicklungsumgebung dlk51 stellt Standard-Funktionen für die Verwaltung des Zielsystems bereit. Über eine Kommandozeile können eigene Kommandos übertragen werden.

so sofort im vollen Umfang einsatzbereit. Angedacht ist die Entwicklung von Systemen mit der 16-bit-CPU Fujitsu 16LX und dem 32-bittigen Hyperstone.

► Entwicklungssoftware sendet Steuerkommandos vom PC aus

Die Entwicklungsumgebung „dlk51“ ist für den Einsatz unter den Betriebs-

Kommando	Beschreibung
File → Setup → ...	init Board und Datentransfer, suche PC-Druckerschnittstelle
File → Program flash → FPGA file ...	programmiere FPGA-Konfigurationsdatei 0 oder 1 im Flash
File → Program flash → CPU file ...	programmiere CPU-Programmdatei 0 – 3 im Flash
File → Program flash → All	programmiere gesamtes Flash-Abbild
File → Save flash ...	schreibe Flash-Abbild auf PC-Festplatte
File → Erase flash → ...	lösche Datei im Flash
FPGA → Config via PC ...	konfiguriere FPGA über PC-Druckerschnittstelle
FPGA → Config via flash	konfiguriere FPGA über Datei im Flash
FPGA → Clear	lösche FPGA-Konfiguration
CPU → Run file ...	starte CPU-Programm 0 – 3
CPU → Disable	halte CPU im Reset-Zustand
Command → Emulate DLCS	konfiguriere FPGA mit „EmuDlc5.bit“, emuliere Xilinx DLCS JTAG-Downloadkabel
Command → Save BASIC RAM ...	sichere MCS BASIC-52-Programm auf PC
Command → Test FPGA I/O	teste FPGA I/O; externer Test-Stecker notwendig
Command → CheckTransferRate	messe Datentransferrate PC/dlk51 und zeige sie an
Command → Commandline ...	öffne Kommandozeilenfenster; nützlich für Test und Entwicklung

Die Entwicklungssoftware, die mit dem „dlk51“ kommuniziert, enthält grundlegende Steuerkommandos, mit denen sich das System konfigurieren lässt.

systemen Windows 9x, NT 4.0, 2000 und dazu kompatiblen ausgelegt (Bild 3). Das Programm dient zur Einrichtung, Verwaltung und zum Test des „dlk“-Systems sowie der Bereitstellung von Funktionen für Test und Datentransfer. Es beherbergt neben den bereits erwähnten Funktionen auch eine leicht für eigene Zwecke erweiterbare Kommandozeileneingabe. So können auch ohne große C++-Kenntnisse kundenspezifische Funktionen eingebaut und getestet werden. Der Quellcode „dlk51“ zeigt den Gebrauch der DLL und den Datenaustausch mit dem PC. Die Kernfunktionen sind in einer DLL in kompilierter Form enthalten. Der Quellcode der DLL ist optional verfügbar. Sämtliche Quellen sind kompatibel zu Microsoft Visual C++ 5.0. Die Tabelle zeigt einen Teil der vom Programm bereitgestellten Funktionen.

Das „dlk“-Konzept stellt eine offene Systemumgebung zur Realisierung der digitalen Kernfunktionen eines Systems dar. Der Quellcode ist vollständig verfügbar. Die grundlegenden Funktionen sind als fertige Blöcke bereits vorhanden. In dieses Gerüst ist eine spezielle Applikation oder eine Anpassung auf einen speziellen Prozessortyp mit voraussagbarem, geringen Aufwand einbaubar. Leistungsfähige und skalierbare Systeme, die aus einer der heute üblichen Systemplattformen wie z.B. Windows NT 4.0 oder 2000 heraus angesprochen werden müssen, sind auf einfache Weise generierbar. Das Kon-

zept eignet sich für die Serien- und Prototyp-Produktion.

Die Einarbeitung in das System ist, durch viele mitgelieferte Beispiele gestützt, innerhalb kurzer Zeit möglich. Für viele Anforderungen bestehen bereits fertige Lösungen in Form FPGA-spezifischer Funktionsblöcke und zugehöriger Quellcodes. Der Aufbau einer für die Benutzer zugänglichen „dlk“-Datenbank ist geplant. Applikationssupport ist verfügbar. *jk*



Dipl.-Ing. (FH) Peter Seng

studierte an der FH Aalen Feinwerktechnik und ist seit 1992 Geschäftsführer der SENG digitale Systeme GmbH. Die Produktpalette umfasste in den letzten Jahren neben Programmiergeräten für Speicher- und Logikbausteine vor allem PC-gestützte, portable Messgeräte. Die Geräte vereint der massive Einsatz von FPGAs und die Konfigurierbarkeit über die parallele Schnittstelle des PC.
 ▶ E-Mail: peter@seng.de