

# Digitale Logikkerne - Technik und Anwendung

Mit FPGA und Prozessor schnell zum lauffähigen Produkt

Ein Konzept zur Entwicklung skalierbarer und einfach programmierbarer, digitaler Kernfunktionalitäten

Von Peter Seng

# SENG

digitale Systeme GmbH

Bestehende Entwicklungs- und Herstellungsverfahren für applikationsspezifische elektronische Geräte eignen sich gut für Märkte mit hohem Produktvolumen. Für Marktsegmente mit kleinem und mittlerem Produktvolumen sind die angebotenen Lösungen oft zu teuer oder anderweitig nicht einsetzbar. Resultierend hieraus werden diese Produkte normalerweise aus PC- oder Microcontroller-Board basierender Standardware hergestellt. Diese ist zwar relativ preiswert, erfüllt jedoch meist die Anforderungen des Produktes nicht optimal. So ist die Entwicklung und das Hinzufügen spezieller Geräteeigenschaften fast immer notwendig. Der Einsatz FPGA basierender Hardware wäre für diese Anwendungen ein durchaus attraktiver Fortschritt, jedoch waren die Einstiegsbarrieren seither hoch, die Entwicklungszeiten und -kosten schwer abzuschätzen.

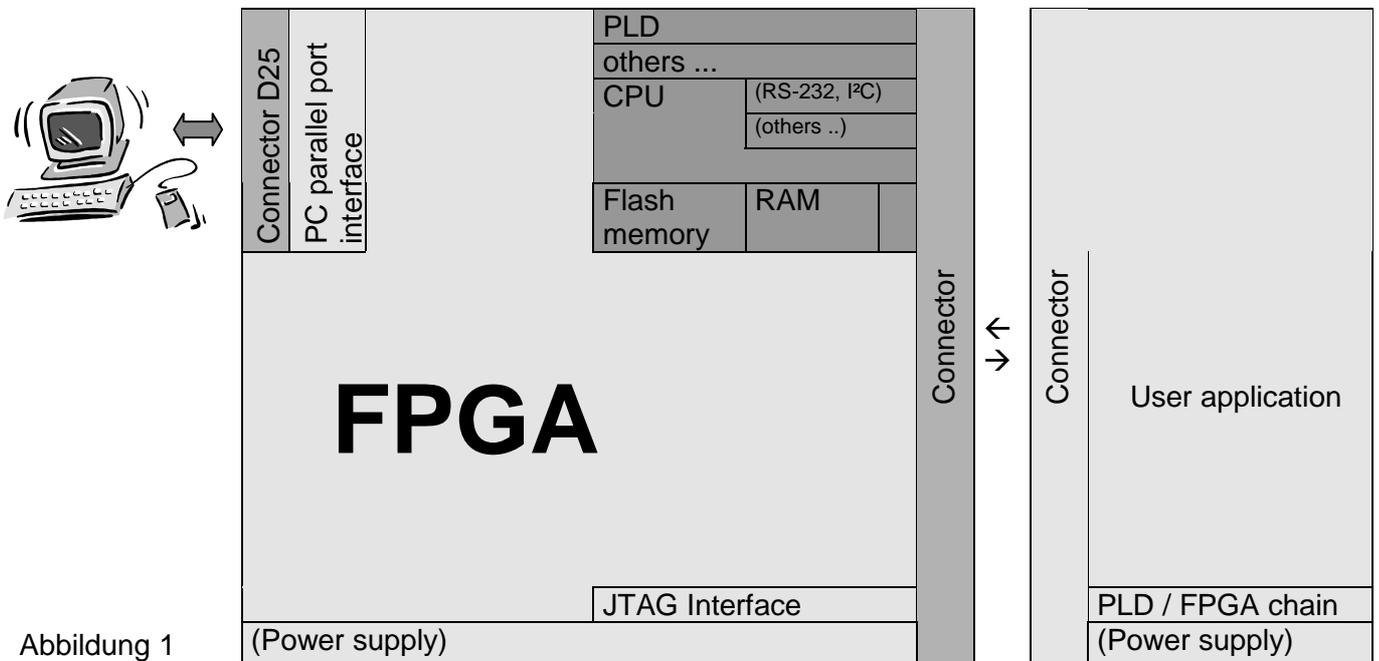


Abbildung 1

Im folgenden möchten wir eine offene Entwicklungsumgebung vorstellen, die es gestattet FPGA basierende Hardware unter Verwendung von Standardbauteilen einfach und unter vorhersagbaren Kosten herzustellen. Das Prinzip nennen wir "digital logic kernel" (dlk). Es umfaßt FPGA + CPU + Speicher + PC Schnittstelle. Es besteht aus Standardbauteilen, FPGA und (C)PLD Quellcode, Software und einem Systemverbindungskonzept. Der "digital logic kernel" wird mit einer Entwicklungsumgebung der Firma SENG digitale Systeme GmbH unter Verwendung von Halbleitern des Herstellers Xilinx und dessen Design Software "ISE" aufgebaut. Nach dem "dlk" Konzept aufgebaute Produkte sind selbständig bootbar, vollständig programmierbar und können mit einem Standard PC Daten austauschen, von diesem verwaltetet oder auch programmiert werden. Aufgrund der integrierten PC-Schnittstelle bietet "dlk" basierende Hardware auch neue Ansatzpunkte und Fortschritte für die Prototypenentwicklung oder den Einstieg in die FPGA Technik. Dieses freiprogrammierbare System bringt bereits ein grundlegendes, lauffähiges Funktionsgerüst mit, hilft damit die Entwicklungszeit zu verkürzen und schafft pflegbare, optimierbare und upgradefähige Systeme. Das Konzept erzwingt keinen Neuentwurf bestehender Lösungen in Form von Hard- und Software, das "dlk" Konzept ist in diese integrierbar. Sämtliche Bestandteile sind im Quellcode verfügbar.

## Typische Problemstellungen

Einige Beispiele bei denen nicht reprogrammierbare Hardware an Ihre Grenzen stößt:

- Eine bewährte Steuerung wird für eine neue Maschinenversion der Überarbeitung unterzogen. Das Konzept benötigt einen weiteren Eingangskanal für Inkrementalgeber. Die verwendete Mini-SPS ist zwar kostspielig in dieser Richtung erweiterbar, jedoch ist eine schnelle und redundante Überwachung von Grenzwerten mit diesem Konzept nicht möglich. Es wird deshalb eine Grenzwertüberwachung mit Microcontroller und etwas Logik "dazugestrickt", deren Einbindung in die SPS zusätzlich erheblichen Aufwand bedeutet.
- Ein Microcontroller gesteuerter Datenlogger wird überarbeitet. Die Erfassrate soll erhöht werden, das Bedienkonzept wird ebenfalls verändert. Nach Erstellen der Hardware wird während der Fertigstellung der Software das Pflichtenheft verändert, um mit einer weiteren Gerätevariante die Anfrage eines potenten Kunden bedienen zu können. Es stellt sich jedoch heraus, daß das alles machbar wäre, wenn man einen FIFO zwischen A/D Wandler und CPU vorgesehen hätte. Die Hardware ist jedoch bereits fertig und ihre Anpassung ist nicht mit den Terminvorstellungen des potentiellen Kunden vereinbar.
- Für ein neues Projekt sollen 3 verschiedene D/A Wandler auf Ihre Leistungsfähigkeit untersucht werden. Um das Ganze zu vereinfachen, soll die Software hierzu auf dem PC entwickelt, die Hardware auf einer Labor-Einsteckkarte aufgebaut werden. Die Störumgebung im PC erweist sich als schlecht abschätzbar, die mehrfache Abänderung der Schaltung ist mit großem Aufwand verbunden. Für die Software-Entwicklung muß auf MS-DOS zurückgegriffen werden, da die Treiber-Software der Einsteckkarte nicht WinNT tauglich ist.

## Vergleich Standardlösung - reprogrammierbare Systeme

3-stufige relative Einteilung ++/+-, Vorteil +, Nachteil -	ohne FPGA	mit FPGA	mit FPGA, dlk basierend
Entwicklungskosten Neuentwicklung (keine FPGA Kenntnisse vorhanden)	+	-	++
Entwicklungskosten Weiterentwicklung	-	+	++
Integrierbarkeit, Platzbedarf Leiterplatte	-	++	++
HF-Eigenschaften	+	++	++
Bauteilekosten **	++	-	+
Wartungskosten	-	+	++
Product-Life-Cycle	-	++	++
Protect from piracy	-	+	+
Flexibilität, erreichbare Marktabdeckung	+	++	++
Differenzierbarkeit der Produktpalette	-	+	++
Prüf und Programmierkosten	-	+	++
Halbleiter-Abhängigkeit	-	+	++
Standard 4 Mbit/sec PC-Interface	-	-	++
PC-Administration	-	-	++
Optimierbarkeit	-	++	++
Kosten Software-Update	+	+	++
Kosten Hardware-Update	-	+	++
Design reusability	-	++	++
SUMME	-	+	++

Tabelle 1

Eine nicht FPGA basierende Lösung kann nur unter der Rubrik "Bauteilekosten" Vorteile verbuchen (Tabelle 1). Dies wird mit vielen Nachteilen erkaufte, die aber für einige Warengruppen in Kauf genommen werden können (z.B. kurzlebige, modische Artikel im Consumer-PC Bereich). Das Design FPGA basierender Systeme schafft für den Produzenten wartbare und somit langlebige Produkte mit hoher Wertschöpfung. Eine "dlk" basierende Lösung ist in der Lage das Vorteilsverhältnis noch stärker in Richtung der FPGA basierenden Lösungen zu verschieben.

**\*\* FPGA's besitzen eine sehr universelle interne Struktur. Die Realisierung einer Funktionalität beansprucht deshalb meist mehr Fläche auf dem Halbleiter (bedeutet höhere Kosten) als bei einem nicht programmierbaren Halbleiter. Die Verlagerung komplexer Standardfunktionalität in das FPGA ist deshalb immer kritisch zu hinterfragen (Beispiele: 8051 CPU oder USB Controller im FPGA).**

Die CPU (oder deren mehrere) kann in einem FPGA basierenden System auf verschiedene Arten integriert werden:

- als Halbleiter auf dem Chip des FPGA (hard-CPU)
- als Netzliste in einem Teil des FPGA (soft-CPU)
- als externer Halbleiter (externe CPU)

3-stufige relative Einteilung ++/+-, Vorteil +, Nachteil -	FPGA mit hard-CPU	FPGA mit soft-CPU	FPGA + externe CPU
Werkzeugkosten	?	?	?
Platzbedarf Leiterplatte	++	++	-
HF-Eigenschaften	++	++	+
Siliziumfläche, Bauteilekosten**	++, ?	-	++
Verfügbare CPU Architekturen	-	-	++
Halbleiter-Abhängigkeit	-	++	+
Erweiterbarkeit CPU	+	++	+
Optimierbarkeit	+	++	+
Flexibilität CPU	-	++	-
Differenzierbarkeit der Produktpalette	++	++	++
Einarbeitungsaufwand	+	+	++
Performance	++	-	+
dlk Kompatibilität	++	++	++
SUMME	?	?	?

Tabelle 2

Die Auswahl der optimalen Lösung (Tabelle 2) kann nur durch die Untersuchung des Bedarfes der Applikation bestimmt werden (\*\*siehe auch Anmerkung vorige Seite).

### Das "dlk" Konzept

Der "digital logic kernel" ist eine einfach anzuwendende Technologie, mit der es möglich ist fast jede Art von Produkt, das eine CPU, programmierbare Logik und Speicher enthält, zu entwickeln, herzustellen und zu programmieren.

Es werden hierzu keinerlei Programmiergeräte oder bereits vorprogrammierte Bauteile benötigt. Eine Grundeigenschaft des "dlk" ist die Möglichkeit selbständig zu booten. Der Zugriff erfolgt über eine integrierte parallele PC Druckerschnittstelle zum Zweck der Administration und des Datenaustausches. Ein Standard-PC mit einer parallelen Druckerschnittstelle ist das einzige Werkzeug das zum Aufbau, der Programmierung und Wartung notwendig ist. Das Konzept ist skalierbar, nur die benötigten Funktionalitäten müssen implementiert werden. (Abbildung 1).

### Bestandteile

Das "dlk" Entwicklungssystem besteht aus FPGA und (C)PLD Hardware Quellcode, Software Quellcode, Software, Schaltplänen des Systemverbindungskonzeptes und einem Demo-Board:

- FPGA interne 8-Bit bi-direktionale Schnittstelle zum PC. Dient der FPGA, (C)PLD und (Flash-)Speicher Programmierung und zum Datenaustausch. Kompatibel zur parallelen Druckerschnittstelle des PC's. Paralleler I/O-Bus mit separatem nWrite und nRead Signal, 256 Adressen direkt adressierbar. Belegt ca. 7% (51 slices) der internen Ressourcen eines xc2s50 FPGA. Die typische Datentransferrate beträgt für Schreiben und Lesen 500 KBytes/sec (4 MBits/sec) im EPP-Modus der Schnittstelle. Hierzu werden Byte orientierte "inbyte()" und "outbyte()" Funktionen bereitgestellt, als Parameter werden direkt Daten und Adressen übergeben (Abbildung 2).
- Treiber und Software, um auf diese Schnittstelle unter verschiedenen PC Betriebssystemen wie Win9x und WinNT 4.0 basierenden (Win2000 etc.) zugreifen zu können.
- Systeminterne, nichtflüchtige (C)PLD basierende Automaten und Speicher, um ein selbständiges Konfigurieren und Booten des Systems zu ermöglichen.
- ein flexibles Systemverbindungskonzept, um diese Technik für verschiedene CPU-, FPGA- und (C)PLD-Typen und -Familien verfügbar zu machen.

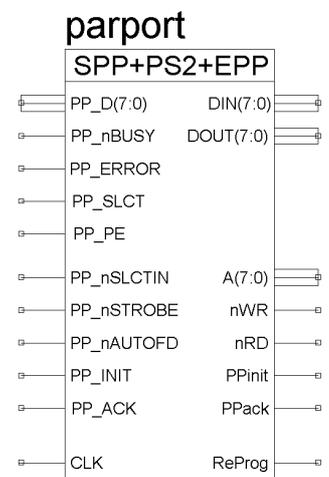


Abbildung 2

- FPGA interne Emulation einer JTAG Schnittstelle, um geräteinterne oder externe Bausteine in einer JTAG Kette zu programmieren. Kompatibel zum Xilinx download Kabel "DLC5" und der "iMPACT" Programmiersoftware. Das System kann also auch direkt mit den Werkzeugen von Xilinx erreicht werden.
- PC Programme zur Systemverwaltung unter Windows, enthält Dynamic Link Library, Applikationsprogramm und Quellcode. Integrierter Kommandozeileninterpreter zur einfachen Integration kundenspezifischer Funktionen. (Abbildung 3 und Tabelle 3).
- Sofort einsetzbares, erweiterbares Demo Board "dlk51" mit xc2s50 oder xc2s100 FPGA; Größe 80x100mm; xc9572xl CPLD; 8032 CPU; 512 KByte Flash-Speicher für 2 FPGA Konfigurationsdateien und 4 unabhängige 64 KByte CPU Programmdateien; i2C Schnittstelle und EEprom; integrierte Stromversorgung; RS-232 und LCD Schnittstelle. Geeignet zur Integration von "soft-CPU's. Die Schaltung ist als komplettes 3V3 Design ausgelegt. (Abbildung 4).
- FPGA Beispiel Quellcode für die Definition der Schnittstelle des 8032 zum FPGA und die Integration weiterer 4 KByte SRAM im FPGA.
- 8032 C Quellcode inclusive LCD, UART, Interrupt und i2c Routinen

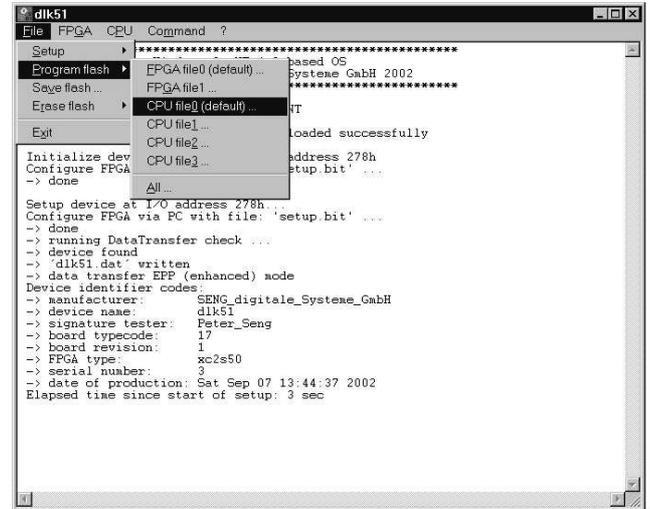


Abbildung 3

(Alle Quellen hierzu sind verfügbar, jedoch im Umfang teilweise abhängig von der erworbenen Lizenz.)

Command	Description
File → Setup → ...	initialize board and data-transfer, search PC parallel printer ports
File → Program flash → FPGA file ...	program on board FPGA file with configuration bitstream data
File → Program flash → CPU file ...	program on board CPU file with program code
File → Program flash → All	program all on board FPGA and CPU files with PC ROM mirror file
File → Save flash ...	save all on board FPGA and CPU files to PC ROM mirror file
File → Erase flash → FPGA file ...	erase FPGA file
File → Erase flash → CPU file ...	erase CPU file
File → Erase flash → All	erase all FPGA and CPU files
File → Exit	save existing setup and exit program
FPGA → Config via PC ...	configure FPGA via PC parallel port interface
FPGA → Config via flash	configure FPGA via on board FPGA file
FPGA → Clear	clear FPGA configuration data
CPU → Run file ...	run CPU program file, active "demo.bit" or "basic52.bit" required
CPU → Disable	keep CPU reset activated, active "demo.bit" or "basic52".bit required
Command → Emulate DLC5	configure FPGA with "EmuDlc5.bit", emulate Xilinx DLC5 JTAG download cable, external available at LIO0..LIO3
Command → Save BASIC RAM ...	save MCS BASIC-52 programm to PC, active "basic52.bit" required
Command → Test FPGA I/O	test FPGA I/O, external test-connector required
Command → CheckTransferRate	check data transfer rate PC ↔ dlk51 and display
Command → Commandline ...	open commandline window, for test and development, basic I/O
? → About dlk51 ...	show program information

Tabelle 3

## Vorteile des "dlk"

- Aufbau der kompletten Kernfunktionalität mit nur vier oder fünf Bausteinen: FPGA, (C)PLD, einem einzigen Flash-Speicher für FPGA Konfigurationsdaten und CPU Programmdateien, hard- oder soft-CPU und einem Standard-Logik-IC der Reihe 74xxx. Zusätzliches RAM kann abhängig vom Bedarf der Applikation und CPU hinzugefügt werden.
- Konfigurieren und booten eines Systems in <35ms (basierend auf xc2s100 FPGA).
- Speicherung mehrerer FPGA Konfigurationsdateien und CPU Programme in einem einzigen Flash-Speicher. Auswahl der Boot-Dateien über ein externes Steuersignal.
- Kompatibel zu fast jeder Art von hard- oder soft-CPU (8-, 16- oder 32-Bit) mit externem Programmspeicher. CPU's mit internem Programmspeicher können ebenfalls verwendet werden, das System bedarf dann jedoch CPU spezifischer Anpassungen (Programmieralgorithmen auf PC Seite und Programmier-Hardware im FPGA).
- Herstellung und Upgrade reprogrammierbarer, skalierbarer und bootbarer Systeme unter Verwendung austauschbarer Standard-Halbleiter ohne den Bedarf jedweden Programmiergerätes oder bereits vorprogrammierter Bauteile.
- Weiterverwendung und Einpflege bereits erworbener Wissensbasis, profitable Weiternutzung bereits vorhandener Technologie.
- Änderung der FPGA Konfiguration "on the fly", ohne Reprogrammierung des Flash-Speichers.
- Steuerung der und Zugriff auf die FPGA interne Logik vom PC, der internen CPU, oder beiden.
- Integrierte schnelle PC-Schnittstelle (4 MBit/sec).
- Falls eine Kommunikation mit dem PC nicht notwendig ist, so werden im FPGA keinerlei interne Ressourcen belegt.
- Falls das Systemkonzept bereits ein FPGA vorsieht, so ist das "dlk" Konzept quasi ohne weitere Kosten integrierbar.
- Die gesamte Software wurde in C/C++ erstellt, kompatibel zu Xilinx ISE (WebPACK).
- Verfügbar als offenes System, in einem sofort einsetzbaren Format und als Quellcode. Vollständig dokumentiert. Sämtliche Quellen sind verfügbar.

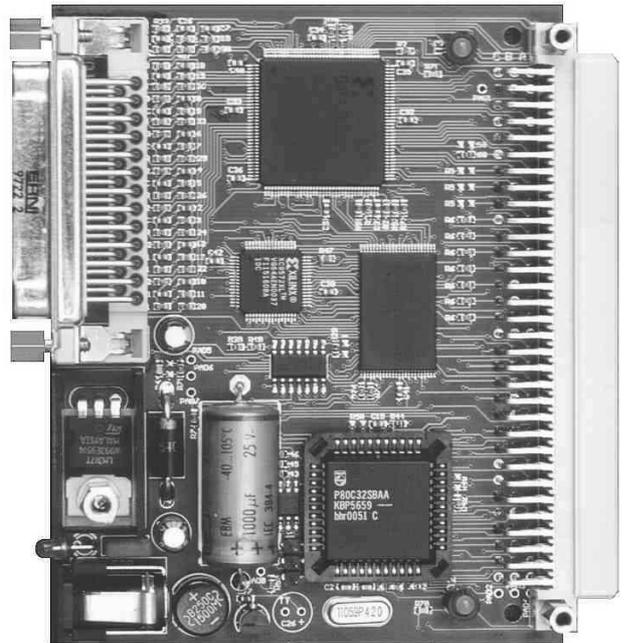


Abbildung 4

## Der "dlk" Entwicklungsprozess

1. Führen Sie Experimente mit dem Demo-Board aus, erforschen Sie das System bis Sie die Funktionalität vollständig verstanden und ein Gefühl für reprogrammierbare Geräte erworben haben.
2. Entwerfen Sie Ihr applikationsspezifisches Board unter Einhaltung des Systemverbindungskonzeptes, Verwendung der "dlk" Quellen, und Ihrer speziellen Vorgaben oder verwenden Sie eines der bereits verfügbaren "dlk" Boards.
3. Arrangieren Sie die gelieferten FPGA, (C)PLD und PC-Software Quellen anhand Ihres speziellen Bedarfs und compilieren Sie diese.
4. Verbinden Sie Ihr applikationsspezifisches Board mit der parallelen PC Druckerschnittstelle und versetzen Sie das Board mit Hilfe der bestehenden "dlk" Software in den JTAG Emulationsmodus.
5. Programmieren Sie das (die) auf dem Board befindliche(n) (C)PLD(s) mit Hilfe des Programmes "iMPACT" der Xilinx "ISE" Software.
6. Entwickeln Sie Ihre applikationsspezifische Logik und CPU Programme und compilieren Sie diese.
7. Laden Sie diese auf das Board.
8. Debuggen Sie Ihre Applikation.
9. Speichern Sie Ihre FPGA Konfigurationsdateien und CPU Programme im Flash-Speicher des Boards und entfernen Sie die Verbindung zur parallelen Schnittstelle des PC's. Ihre Applikation ist nun eigenständig lauffähig.
10. Zum Zwecke der Systemverwaltung oder des Datenaustausches können Sie die Verbindung zum PC jederzeit wieder herstellen.

## **Weitere besondere Eigenschaften**

Das "dlk" System eignet sich für jede Art von CPU, eingeschlossen sogenannter "soft-CPU's" - 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 den Zielprozessor des Systems übertragen werden. Die Steuerung des Systems komplett 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's 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. 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 5V toleranter Bausteine ist jedoch in vielen Applikationen sicherlich von Vorteil.

## **Anwendungsgebiete**

Hier erstreckt sich ein nicht übersehbares Feld von Möglichkeiten. Der Einsatz als Entwicklungsplattform, die Implementierung digitaler Kernfunktionalitäten für jede Art von Applikation, die schnelle Prototypenentwicklung, 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 sogenannter Chamäleon-Systeme sei ebenfalls verwiesen. Falls also zum Zeitpunkt des Entwurfes das künftig zu erfüllende Lastenheft noch nicht vollständig bekannt ist oder die Applikation oder Normung Veränderungen unterworfen sein wird, so bietet das "dlk" Konzept ebenfalls die geforderte Flexibilität und Wandelbarkeit.

## **Zusammenfassung**

Das "dlk" Konzept ist eine offenes System das Standardwerkzeuge und bereits vorgefertigte Elemente benutzt und bereitstellt und hiermit die Entwicklung und die Herstellung digitaler reprogrammierbarer Produkte schnell, kostengünstig und mit vorhersagbarem Aufwand ermöglicht. Das System eignet sich vor allem für kleine und mittlere Produktvolumen wie auch für die Prototypenerstellung. Es ist kompatibel zu Xilinx ISE und den Windows Betriebssystemen. Das System beinhaltet mehrere einsetzbare Beispielapplikationen und ein voll dokumentiertes Demo Board um einen schnellen Einstieg zu ermöglichen. Der Quellcode ist vollständig verfügbar. Applikationssupport und Entwicklung wird angeboten.

## **Kontakt**

SENG digitale Systeme GmbH  
Im Bruckwasen 35  
D 73037 Göppingen  
Germany  
tel +7161-75245  
fax +7161-72965  
eMail [info@seng.de](mailto:info@seng.de)  
net <http://www.seng.de>  
embedded world 2003, Halle 12.0, Stand 12-401

## FAQ zum Thema "dlk"

*Warum wird der PC Parallel-Port verwendet, und nicht USB oder ...?*

Der PC Parallel-Port ist die einzige externe Schnittstelle des PC's die es gestattet ein Gerät anzusteuern, das keinen oder noch keinen lauffähigen Prozessor oder Automaten enthält. Diese Schnittstelle dient primär Entwicklungs- und Verwaltungszwecken, sie kann natürlich auch im fertigen Produkt genutzt werden. Wir gehen davon aus, daß das fertige Produkt evtl. noch weitere Schnittstellen haben wird. Diese Schnittstelle ist jedoch die erste die in Betrieb gehen und durch die das Produkt zum Leben erweckt werden kann, ganz unabhängig von irgendwelchen speziellen Halbleitern und Programmiergeräten, und ohne die notwendige Programmierung eines sehr aufwendig zu implementierenden Protokolls. Zudem verteuert die Schnittstelle das Produkt nur um wenige Cent.

*Muß der Parallel-Port Steckverbinder oder die gesamte Schnittstelle immer mit auf das "dlk" Board?*

Nein, im einfachsten Falle genügen auch einige Pads auf der Platine für einen Nadeladapter oder ein Miniatursteckverbinder. Benötigt wird dann aber in beiden Fällen ein externer, passiver Adapter zum PC.

*Ist "dlk" nur mit Xilinx Halbleitern machbar?*

Sicher ist es auch mit anderen Halbleitern machbar, aber in der Anfangszeit der FPGA's gab es keine anderen und wir sahen über all die Jahre auch keinen Grund den Hersteller zu wechseln. Für eine Auftragsentwicklung setzen wir natürlich nach einer Machbarkeitsprüfung auch gerne einen anderen Hersteller auf das Board, falls der Kunde die Entwicklung bezahlt.

*Ist mit FPGA Hardware ein robustes Produkt herstellbar?*

Unsere Erfahrung sagt ja. Wir haben diese Technik in portablen Programmiergeräten und Meßgeräten, die vielfach im Fahrzeug eingesetzt werden, seit Jahren erfolgreich im Einsatz.

*Ist dies eine Abkehr von den sogenannten Standardmodulen?*

Ja und nein.

Nein, schauen Sie auf z.B. unser "dlk51" Modul. Es ist ein Standardmodul - nur eben sehr viel flexibler und leichter zu handhaben, als das, was wir seither gewohnt waren. Weitere Standardmodule werden folgen. Unflexible, heute übliche Module können leicht zu "dlk" Modulen weiterentwickelt werden. Der Aufwand hierzu ist, unsere Entwicklungswerkzeuge vorausgesetzt, minimal.

Ja, weil im FPGA Bereich der Module applikationsspezifische Hardware darstellbar ist, und ein Modul obwohl es gleich aussieht wie ein anderes, trotzdem vollkommen verschiedene Funktionen beinhalten kann.

Eine Produkt besteht nicht nur aus dem Digitalteil, es gibt auch immer noch das "Drumherum", und das ist meist das Eigentliche. Es muß also für jedes Produkt immer eine Leiterplatte erstellt und auch geprüft werden (Funktion, Streß, EMV). Wir denken, daß es bei den heutigen Entwicklungswerkzeugen und Ansprüchen oft einfacher und preiswerter ist, ein Produkt auf möglichst wenige Module zu verteilen. Oft ist es das Beste, alles auf eine Platine (mit reprogrammierbarem Kern) zu setzen. Der Markt für traditionelle Standardmodule wird enger.

*Warum gibt es als erste Implementierung ein 8051 kompatibles Demo Board ?*

Der 8032 wurde ausgewählt, da es sich wohl um den populärsten Prozessor handelt und eine Vielzahl von Entwicklungswerkzeugen und Programmquellen frei verfügbar ist. Ein C-Compiler (SDCC, unter GPL Lizenz) und C Programmbeispiele werden mitgeliefert. Das System ist also sofort im vollen Umfang einsatzbereit.

*Gibt es weitere Demo-Boards ?*

Angedacht ist die Entwicklung von Systemen mit der 16-Bit Fujitsu 16LX und der 32-Bit hyperstone CPU.