

---

## **Programmierbarer Taktgenerator mit CoolRunner® von Xilinx**

Gunther Zielosko, Heiko Grimm

### **1. Die Aufgabe**

Wie schon im vorigen Bericht beschrieben, wollen wir Anwendungen für den BASIC-Tiger® entwickeln, die dessen Intelligenz mit schneller Hardware auf der Basis komplexer programmierbarer Logikbausteine (CPLD's) koppeln. Als Basis-Schaltkreise haben wir die CoolRunner®-Familie von Xilinx gewählt, die neuerdings als CoolRunner®2 auch für Anwendungen bis 300 MHz (!) verfügbar sind.

Was genau soll nun unser programmierbarer Taktgenerator leisten?

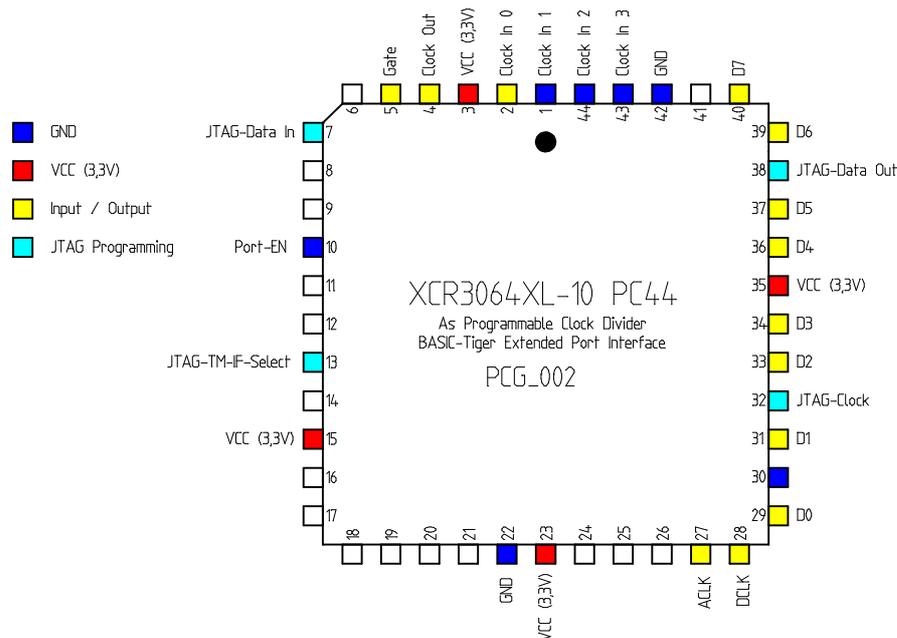
In fast jeder Elektronikwerkstatt wird ein Taktgenerator gebraucht. Am besten wäre es, wenn er quartzgenau arbeiten würde, dennoch aber in der Frequenz einstellbar. Eigentlich ein Widerspruch – denn quartzgenau heißt, daß er mit einer festen Frequenz arbeitet. Wer schon von PLL's (Phase Lock Loop – phasengeregelte Frequenzeinstellung) gehört hat, weiß, daß es da Tricks gibt. So schön das funktioniert, wir wollen einen anderen Weg gehen, denn eine genaue Frequenz bekommt man nur durch Teilung der quartzgenauen Grundfrequenz. PLL's zeigen nämlich wegen ihres Regelverhaltens geringfügige Schwankungen in der Frequenz (Jitter). In der Regel werden in der Digitaltechnik Frequenzen durch Flip-Flops heruntergeteilt, d.h aus z.B. 10 MHz Eingangsfrequenz wird nach einem Flip-Flop die halbe Frequenz, also 5 MHz. Nach einem weiteren Flip-Flop werden 2,5 MHz ausgegeben usw. Zwar sehr elegant, aber für viele Zwecke nicht variabel genug. Bessere Resultate bekommt man mit einem programmierbaren Teiler, der die Eingangsfrequenz durch eine (nahezu) beliebige Zahl teilen kann. Die Realisierung erfolgt dadurch, daß man einen Zähler aufbaut, der die Eingangsimpulse zählt, das Ergebnis an einen Bitvergleichler weitergibt, der es mit einem vorgegebenen Wert vergleicht. Ist der Zählerstand erreicht, der dem vorgegebenen Wert entspricht, wird der Zähler rückgesetzt und beginnt erneut zu zählen. So wird erreicht, daß er beispielsweise genau nach vorgegebenen 1000 Eingangstakten rückgesetzt wird. Der Rücksetzimpuls wird ausgewertet und als Ausgangsfrequenz ausgegeben. Damit das Ganze richtig komfortabel wird, sollte der Zähler recht schnell sein (hohe Ausgangsfrequenzen), viele Stufen aufweisen (feine Frequenzabstufungen) und ein elegantes Interface beinhalten (einfache Ansteuerung durch den BASIC-Tiger®).

Unsere Vorgaben lauten:

- Eingangsfrequenz möglichst von 0 bis 100 MHz,
- 24 Bit-Teiler (16,8 Millionen Stufen !),
- als Interface für den BASIC-Tiger® wollen wir das erweiterte Eport-System nutzen, d.h. der zu entwickelnde Baustein muß die komplette Logik von 3 Eports zu je 8 Ausgängen mit enthalten (im Plug-and-Play-Lab sind das 4 Stück 74HC377 und 1 Stück 74HC138).

## 2. Die Lösung mit XCR3064XL im 44-poligen PLCC-Gehäuse

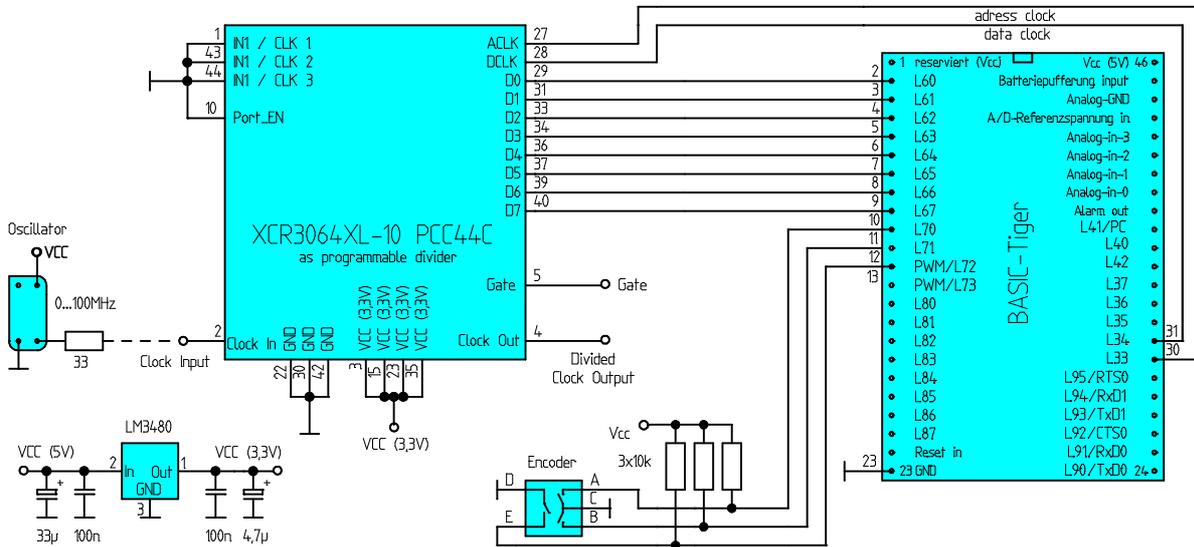
Die oben gestellte Aufgabe wird durch einen einzigen Baustein XCR3064XL im 44-poligen PLCC-Gehäuse gelöst. Die Anschlußbelegung zeigt Bild 1, Bild 2 den Anschluß an den BASIC-Tiger®, wie man sieht, eine sehr einfache Sache.



*Bild 1 XCR3064XL im 44-poligen PLCC-Gehäuse als programmierbarer Teiler (PCG\_002)*

Die hellblau gekennzeichneten JTAG-Pins dienen zur Programmierung des XCR3064XL-10 PC44 und werden in der Anwenderschaltung ebenso wie die restlichen weiß gekennzeichneten Pins offengelassen. Die Clock-Inputs 1, 2 und 3 sowie das Pin Port\_EN haben keinen Pullup-Widerstand und müssen auf GND gelegt werden. Die VCC (3,3V)-Pins werden aus einer eigenen Regelschaltung (z.B. LM34801M3\_3V) versorgt und gut gegen Masse abgeblockt. Immerhin soll unser Teiler mit maximal 100 MHz arbeiten, da braucht auch ein „CoolRunner®“ dynamisch schon einigen Strom.

Obwohl der CoolRunner® mit 3,3 V arbeitet, toleriert er auch 5 V – Eingangssignale, so daß einer direkten Kopplung mit dem BASIC-Tiger® nichts entgegensteht. Umgekehrt akzeptieren die meisten Logikfamilien auch die 3,3 V – Pegel des CoolRunners®.



*Bild 2 Anbindung an den BASIC-Tiger®*

Zur Funktion:

Das integrierte Interface benutzt soft- und hardwareseitig das Eport-System des BASIC-Tigers® zum Setzen des Teilerfaktors, also den Port 6 als Datenbus sowie die beiden Signale ACLK und DCLK. Mit dieser Technik kommt man ohne weitere Hardware und minimaler Verdrahtung aus. Über drei Adressen des Eport-Systems wird das 24-Bit-Datenwort nacheinander in drei 8-Bit-Register des CoolRunners® geschrieben. Als festverdrahtete Adressen für die OUT-Befehle der drei 8-Bit-Worte dienen die Adressen 10h (niederwertiger Teil), 11h (mittlerer Teil) und 12h (höherwertiger Teil), diese Adressen sowie 13h sind also für andere Anwendungen nicht mehr verfügbar, die Adresse 13h (Bit 0) wird als RESET für den Teiler genutzt. Solange Bit 0 auf Adresse 13h „1“ gesetzt ist, bleibt der Teiler im Ruhezustand und am Ausgang passiert nichts.

Die Teilerkette mit 24 Teilerstufen beginnt am Eingang „Clock In“ (Pin 2) und endet am Ausgang „Clock Out“ (Pin 4). Die Eingangsfrequenz  $F_i$  wird immer durch den Faktor 2 geteilt, so daß maximal die halbierte Eingangsfrequenz am Ausgang erscheint. Dies liegt an dem letzten Flip-Flop, das aus dem inneren kurzen Reset-Impuls ein symmetrisches Ausgangssignal macht. Der Teilerfaktor „0“ ist nicht erlaubt, beim Teilerfaktor 2 wird die halbe Eingangsfrequenz ausgegeben, beim Teilerfaktor 4 wird die Eingangsfrequenz durch 4, beim Teilerfaktor 6 durch 6 geteilt usw. Die Ausgangsfrequenz  $F_o$  leitet sich nach folgender Formel aus der Eingangsfrequenz ab:

$$F_o = F_i / T \quad (T = \text{Teilerfaktor, } 2, 4, 6, 8, \dots, 2 * 16,7 \text{ Millionen})$$

Beispiel: Eingangsfrequenz  $F_i = 4,915200$  MHz

Teilerfaktor T	Ausgangsfrequenz $F_o$
2	2,457600 MHz
4	1,228800 MHz
6	819200 MHz
20	245760 kHz
1 000	4915,2 kHz
1 000 000	4,9152 Hz

Bedenken Sie, daß ein Teilerfaktor bis  $2 \cdot 16,8$  Millionen möglich ist. Dadurch und durch die theoretische Eingangsfrequenz bis 100 MHz lassen sich praktisch beliebige Frequenzen einstellen, wenn Sie Zeit haben, bis hin zu Jahren! Ein Wort noch zu der real erreichbaren Eingangsfrequenz. Die Autoren haben mit der 10ns Version des CoolRunners® immerhin bis zu 40 MHz erreicht, obwohl der Baustein mit seinem 24-stufigen Zähler, dem Bitvergleicher (in Echtzeit) und dem nicht ganz einfachen Interface nahezu „voll“ ist. Das bedeutet für die innere „Verdrahtung“ schon Kompromisse bezüglich der Signalverläufe und damit Einschränkungen der Geschwindigkeit. Mit einem 6ns-Exemplar oder einem Typ mit mehr Makrozellen und Registern sollten 100 MHz realistisch sein.

Die Ausgangsfrequenz hat immer ein Tastverhältnis von 1:1, d.h. die Schwingung ist symmetrisch.

Der Gate-Eingang (Pin 5) dient zum externen Ein- bzw. Ausschalten des Ausgangssignales. Bei offenem Eingang oder einer logischen „1“ kommt am Ausgang die eingestellte Frequenz heraus, bei logischer „0“ oder Verbindung mit Masse liegt der Ausgang fest auf GND. Dies ist eine Möglichkeit, die Ausgangsfrequenz zu tasten, d.h. sie im Takt einer Information ein- und auszuschalten. Solch eine Funktion wird z.B. bei Fernbedienungen gebraucht, wo Infrarot-LED's bei etwa 35 kHz mit kodierte Befehlen getastet werden. Diese Tastung kann von irgendwelchen Fremdgeräten, aber natürlich auch vom BASIC-Tiger® aus erfolgen – tolle Möglichkeiten für den Einsatz unseres programmierbaren Frequenzgenerators.

Der an den Pins L70 und L71 angeschlossene Encoder mit seinem Tastenkontakt an L72 dient zur Einstellung des Teilerfaktors im Mustergerät und kann bei anderweitiger Frequenzeinstellung einfach weggelassen werden. Im Programm PCG\_002.TIG stellen wir mit diesem Encoder die Frequenz in weiten Bereichen ein.

### 3. Das BASIC-Tiger®-Programm PCG\_002.TIG

Der Name PCG\_002.TIG steht für „Programmable Clock Generator“. Das Programm realisiert mit der beschriebenen Hardware einen komfortablen programmierbaren Taktgenerator. Die Quarz- bzw. Eingangsfrequenz wird als Variable  $F_i$  bereits im Programm eingegeben. Wer möchte, kann auch dies mit einer Eingabefunktion (z.B. mit dem Encoder!) realisieren. Danach folgt ein Programmteil zur dynamischen Abfrage des Encoders, d.h. bei schneller Drehung werden sehr weite „Sprünge“ realisiert, während bei langsamer Drehung

ganz feinfühlig nur Einerschritte ausgeführt werden. Das Ergebnis wird multipliziert mit 2 als Teilerfaktor T benutzt (Bereich 2...2\*16,7 Millionen). Daraus errechnet das Programm die Ausgangsfrequenz und die Periodendauer. Angezeigt werden:

die Eingangsfrequenz (Fi),  
der Teilerfaktor (T),  
die Ausgangsfrequenz (Fo),  
die Periodendauer (P).

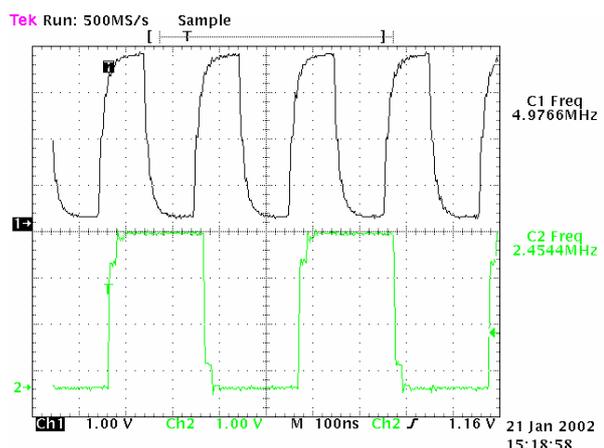
Die Eingangsfrequenz Fi wird in Hertz angezeigt, die Ausgangsfrequenz Fo ebenfalls. Hier gibt es allerdings Dezimalstellen (Komma), da die Ausgangsfrequenz auch sehr klein werden kann. Ähnliches gilt für die Periodendauer P, die von ns bis zu vielen Sekunden dauern kann. Die „Multianzeige“ mit Fi, T, Fo und P ist beim Einstellen sehr hilfreich, da man den Blick auf den gewünschten Wert konzentrieren und dann möglichst genau auf den Zielwert einstellen kann.

Noch ein paar Hinweise zur Frequenzeinstellung. Um keine undefinierten Zustände der Zählerkette zuzulassen, ist es sinnvoll, den Teiler während der Einstellphase „ruhigzustellen“. Wir tun das im BASIC-Tiger<sup>®</sup>-Programm PCG\_002.TIG durch einen Reset über die Eport-Adresse 13h mit einer „1“. Dann erst werden nacheinander die drei Bytes des 24-Bit-Wortes über die Adressen 10h, 11h und 12h ausgegeben. Erst danach wird der Reset-Zustand aufgehoben.

Übrigens läßt sich mit dem BASIC-Tiger<sup>®</sup>, dem Teiler mit CoolRunner<sup>®</sup> und einem geeigneten BASIC-Tiger<sup>®</sup>-Programm auch ein exzellenter Frequenzmesser realisieren. Dadurch, daß Sie für die Messung einen beliebigen Teilerfaktor einstellen können, lassen sich Frequenzen in einem weiten Bereich sehr genau messen.



*Bild 3* Displayausgabe bei einer Quarzfrequenz von 4,915200 MHz und Faktor 2 (maximale Frequenz)



*Bild 4* Eingangssignal (schwarz) und Ausgangssignal (grün) bei Teilerfaktor 2



Bild 5 ...und bei Faktor 10

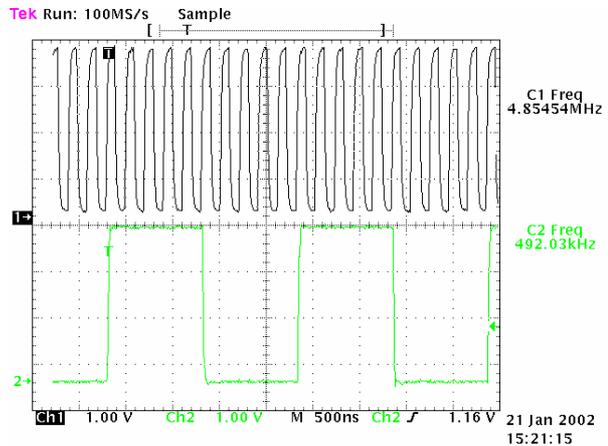


Bild 6 Eingangs- (schwarz) und Ausgangs-  
signal (grün) bei Teilerfaktor 10

Das Programm PCG\_002.TIG ist natürlich nur ein einfach gehaltenes Beispiel, das auf die Bedürfnisse des Nutzers angepaßt werden kann. So ist neben der erwähnten Eingabe der Eingangsfrequenz auf einfachste Weise auch deren indirekte Messung möglich. Sowohl die im BASIC-Tiger<sup>®</sup>-Handbuch als auch die in einigen Applikationsberichten dieser Reihe dargestellten Möglichkeiten der Frequenzmessung können eingesetzt werden. Sie müssen lediglich den Taktausgang so einstellen, daß die Ausgangsfrequenz für den BASIC-Tiger<sup>®</sup> meßbar wird, also einige kHz bis einige Hz. Dann wird diese Frequenz ermittelt und die Eingangsfrequenz über den Teilerfaktor rückgerechnet. Als letzter Schritt dieser Aktion wird der Wert in die Variable Fi übernommen. Achten Sie aber darauf, daß die Verbindung zum BASIC-Tiger<sup>®</sup> nicht aus Versehen mit vielen MHz „gestreßt“ wird.

Zum Abschluß im Bild 7 noch ein Musteraufbau der Autoren für das Minilab. Wie man sieht, ist der Hardwareaufwand für die Aufgabenstellung minimal. Wahlweise kann die Eingangsfrequenz durch einen Quarzgenerator oder über eine HF-Buchse eingegeben werden. Rechts ist der Encoder zu erkennen.

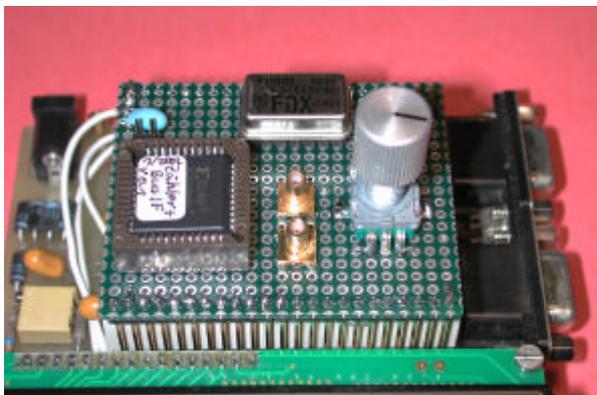


Bild 7 Musteraufbau mit Lochraster-  
Leiterplatte für das Minilab

#### 4. Eine stand-alone Version des programmierbaren Teilers

Im Verlauf der Entwicklung entstand zunächst eine Teilerversion, bei der die 24-Bit-Information direkt, also ohne BASIC-Tiger<sup>®</sup>-Interface angelegt wurde. Diese Version kann beispielsweise per DIL-Schalter, Jumper oder ein anderes Mikrorechnersystem programmiert werden. Wegen der einfacheren inneren Verdrahtung ist diese Variante zudem erheblich schneller. Hier konnten auch in der 10ns-Version des CoolRunners<sup>®</sup> 100 MHz erreicht werden. Bild 8 zeigt die Pinbelegung dieser Version und Bild 9 einen Laboraufbau der Autoren. Die Eingangsfrequenz wird an Pin 2 angelegt, die geteilte Frequenz wird am Pin 41 ausgegeben. RESET ist high-aktiv und die 24 Bit für den Teilerfaktor werden statisch an die Eingänge In 0 bis In 23 angelegt. Die diesem Bericht beiliegende Datei PCG\_001.JED beinhaltet eine ausgetestete Lösung.

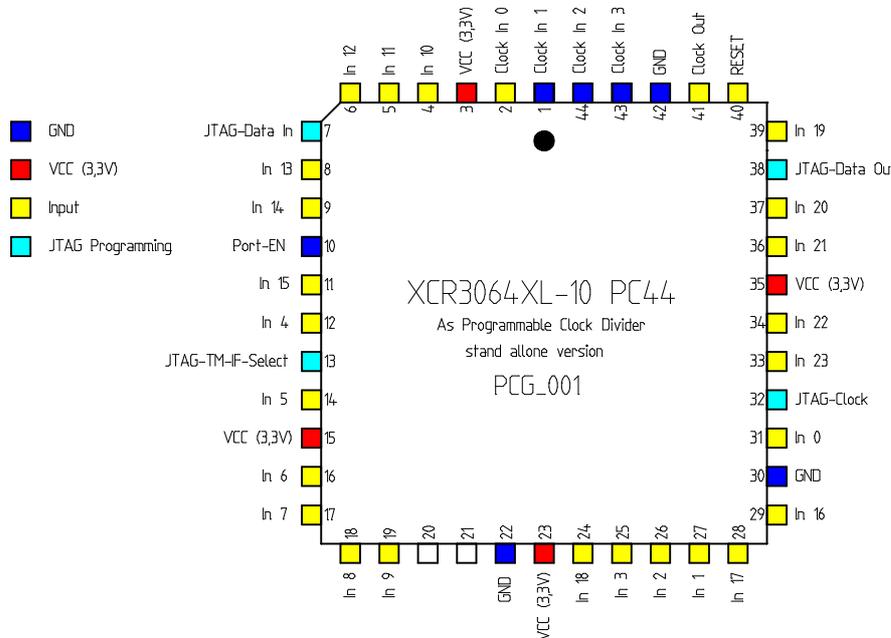
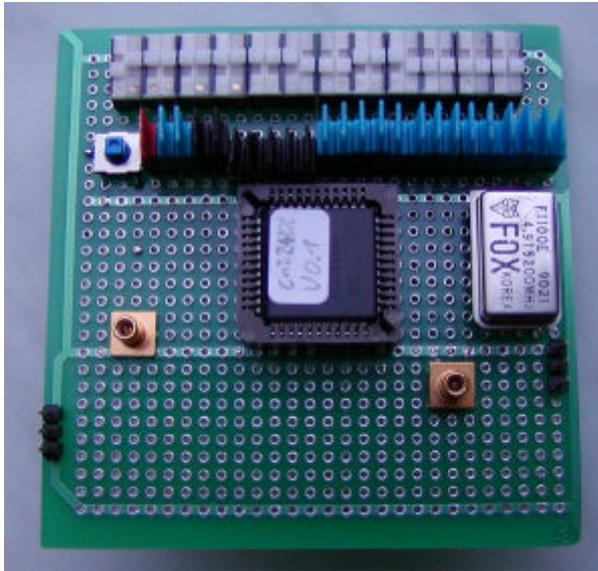


Bild 8 Pinbelegung der stand alone Version



*Bild 9 Teiler ohne Interface*

## **5. Wie kommt man an die XCR3064XL als programmierbare Teiler**

Eine fertige Lösung bieten wir für Kenner des XILINX-CoolRunner®-Systems mit den Dateien PCG\_001.JED und PCG\_002.JED an, die diesem Applikationsbericht beigelegt werden. Etwas schwieriger wird es für Neulinge auf diesem Gebiet. Sie haben die Möglichkeit, sich mit den Informationen und der Software von den XILINX-Webseiten sowie dem dort ebenfalls angebotenen Programmieradapter selbst „durchzubeißen“. Andererseits wird der Autor mit Unterstützung der Firma Wilke bei Interesse ggf. fertige Platinen, Programmieradapter und/oder fertig programmierte Bausteine anbieten.