

# Cloning obsoleter Baugruppen mittels programmierbarer Logikbausteine

Dipl. Inform. Rainer Buchty, Technische Universität München  
eMail: buchty@in.tum.de

CPLD Entwicklerforum 1999

*In der Vergangenheit bot die Industrie eine Vielzahl hochintegrierter Bausteine an, welche auf bestimmte Einsatzgebiete zugeschnitten waren. Beispiele hierfür sind Ein-/Ausgabebausteine, DRAM-Controller wvm. Im Zuge immer höherer Integrierung wurden diese Funktionen Bestandteil des Chipsets moderner PC-Motherboards, die meist noch aus der PC/XT-Zeit stammenden Peripheriebausteine wurden hierdurch quasi obsolet.*

## 1 Einleitung

Die Integration von vielen Funktionen in einige wenige Bausteine ist vor dem Hintergrund der Kostenreduzierung sicherlich begrüßenswert, jedoch hat sie auch dazu geführt, daß beliebte Peripheriebausteine vom Markt verschwunden sind. Für den Hobbyelektroniker stehen diese Bausteine sicherlich noch in ausreichender Zahl zur Verfügung, für die gewerbliche Massenproduktion indes verbietet sich die Verwendung

obsoleter Bausteine.

Doch was tun, wenn solche Bausteine in einem bestehenden Design eingesetzt werden? Üblicherweise wird nun der Designer die betreffende Baugruppe einem Redesign unterziehen, die obsoleten Bausteine werden durch noch erhältliche ersetzt und die Baugruppe ggf. hierfür angepaßt. Im Falle älterer Designs wird dieses oftmals auch einem Shrink unterzogen, Logikfunktionen werden in programmierbare Logikbausteine (PLDs) zusammengefaßt. Aber auch entsprechend jüngere Designs, bei denen bereits PLDs eingesetzt wurden, können von einem Redesign profitieren, denn auch diese Bausteine unterliegen einem Preisverfall bzw. warten bei gleichen Kosten mit besserer Leistung auf.

Es gibt allerdings auch den Fall, da das Design nicht angetastet werden darf oder kann. Dies ist beispielsweise im Reparaturbereich der Fall, hier kann die Baugruppe selbst nicht verändert werden. Bestenfalls möglich sind

hier einfache Anpassungen, die es ermöglichen, funktionskompatible Bausteine anstatt der obsoleten Bausteine einzusetzen. Doch meist gibt es keine wirkliche Alternative zu den ursprünglich verwendeten ICs - hier muß man sich bei der Reparatur ständig schwindender Abverkäufe oder recycelter Bausteine bedienen.

## 2 Chipcloning – eine Alternative

Heutige PLDs sind ungleich komplexer als die ersten PLAs und PALs der 80er Jahre. Moderne CPLDs warten mit über 512 Makrozellen und mehr als 20.000 Gatteräquivalenten auf und FPGAs mit 100.000 Gatteräquivalenten sind längst nicht mehr die Ausnahme. Auch die Entwicklungssoftware ist mittlerweile gereift und erlaubt die Erstellung von Logikdesigns in Hochsprachen wie VHDL und Verilog, jedoch ohne die langen Übersetzungs- und Fittingzeiten (Place and Route), wie sie noch vor wenigen Jahren die Regel war.

Dies erlaubt nun, basierend auf den bekannten Funktionen der verwendeten Bausteine Verhaltensmodelle zu erstellen. Werden diese Verhaltensmodelle in einer entsprechenden Art und Weise verfaßt, so können sie auch ohne weitere Änderung zur Synthese verwendet, d.h. die Bausteinfunktionen in ein CPLD bzw. FPGA (im folgenden unter dem Begriff PLD vereint) abgebildet werden. Handelt es sich um eine 1:1-Abbildung der Funktionen,

so spricht man auch vom Cloning eines Bausteins.

## 3 Grundsätzliche Überlegungen

So mächtig moderne PLDs sind, so bleibt ihr Einsatzgebiet strikt auf digitale Bereiche beschränkt. Dies gilt es stets zu beachten, will man einen nicht mehr erhältlichen Schaltkreis mittels PLDs nachempfinden. In manchen Fällen vereint ein solcher Schaltkreis in sich nämlich nicht nur digitale sondern auch analoge Funktionen. Ein Beispiel hierfür sind Videochips: Zwar ist die Bilderzeugung letztendlich digital (Zähler, Vergleicher, Schieberegister...), doch enthalten diese Chips eine analoge Aufbereitung dieser Signale, um (historisch) Y/C- oder FBAS-Signale auszugeben. Ein anderes Beispiel sind integrierte Soundchips, die nicht nur einen oder mehrere Oszillatoren und Verstärkerstufen enthalten sondern auch analoge Filter oder gar A/D-Wandler, wie es noch vor wenigen Jahren durchaus üblich war. Derlei Funktionsgruppen lassen sich nur schwer bis gar nicht durch PLDs emulieren und bedürfen zusätzlicher externer Komponenten.

Eine weitere, sehr wichtige Überlegung ist, ob die Funktionen des zu klonenden Bausteins 1:1 übernommen werden müssen, oder ob ggf. nur ein Bruchteil dessen Fähigkeiten im bestehenden Design benötigt werden. In diesem Fall müssen naheliegenderweise auch nur die

tatsächlich benötigten Funktionen integriert werden. Ein Beispiel hierfür wäre der Ein-/Ausgabebaustein 8255, bei dem sehr oft nur die Basiskonfiguration verwendet wird und die anderen, teilweise etwas schwieriger zu implementierenden, Modi gar nicht zum Einsatz kommen.

## 4 Replacement oder Redesign?

Die Fragestellung nach der nachzuempfindenden Funktionalität stellt sich gleichermaßen beim Replacement (Ersetzung eines Bausteins durch einen PLD-Klon) wie auch beim Redesign (Neukonzeptionierung einer Baugruppe). Im zweiten Fall hat man es naturgemäß leichter, da einem hier meist alle Unterlagen bezüglich der benutzen Bausteinfunktionen zur Verfügung stehen, woraus die tatsächlich benötigte Funktionalität der zu ersetzenden Bausteine hervorgeht. Auch bietet das Redesign die Möglichkeit, mehrere Bausteine bzw. deren Funktionen in ein PLD zusammenzufassen, um somit die Herstellungskosten zu senken.

Wird jedoch die gesamte Funktionalität genutzt, so benötigt man eine genaue Abbildung der Bausteinfunktionen – ein Punkt, der insbesondere beim Replacement von Bausteinen zum Tragen kommt. Gerade hier muß man im Sinne maximaler Kompatibilität alle Bausteinfunktionen abbilden, nach Möglichkeit sogar eventuelle „illegale“, d.h. vom Hersteller zwar nicht spezifizier-

te, aber aufgrund der Fertigungsmethode o.ä. dennoch möglichen Funktionen. Ein Beispiel hierfür wäre der Mikroprozessor 6502 von Rockwell: In der ursprünglichen NMOS-Variante verfügte dieser Prozessor über eine Vielzahl von sogenannten illegalen Opcodes, die aufgrund der langjährigen unveränderten Verfügbarkeit auch Einzug in den Programmcode fanden (Dies geschah aus Geschwindigkeitsgründen, da gewisse illegale Opcodes die parallele Modifikation mehrerer Register in nur einem Taktzyklus erlaubten. Möglich war dies aufgrund der festverdrahteten Logik des 6502.) Heute existierende 6502-Cores berücksichtigen indes nur den von Rockwell spezifizierten Befehlssatz und sind deshalb für den Einsatz in solchen (Software-)Umgebungen nicht direkt einsetzbar.

Um bei diesem Beispiel zu bleiben – auch das externe Verhalten eines Bausteins muß exakt nachempfunden werden: Es hilft nicht, nur die prinzipielle Funktionsweise eines Bausteins nachzubilden, auch die Art und Weise, wie er mit anderen Bausteinen kommuniziert, muß berücksichtigt werden. In diesem Falle ist somit nicht nur die Prozessorfunktion nachzubilden sondern auch das Busprotokoll, will man den PLD-Klon als vollwertigen Ersatz verwenden, auch wenn sich das Busprotokoll durch geeignete Implementierung zweifelsohne beschleunigen ließe. Dies würde dann aber zu einer Veränderung dieses Protokolls führen, so daß ein solcher Baustein nicht in allen (Hardware-)Umgebungen einsetzbar wäre.

## 5 Abschätzungen und Modellierung

Die Auswahl des PLDs sollte wohlüberlegt sein. Wie bei jedem anderen Design gilt es natürlich auch beim Chipcloning die Anforderungen an den PLD sorgfältig zu analysieren, um nicht später in der Fitting-Phase an die Grenzen des PLDs zu stoßen. Bei der Implementierung von Funktionen komplexer Peripheriebausteine z.B. ergeben sich aber noch zusätzliche Fallen: Oftmals enthalten diese versteckte Register in Form interner Zähler, Ereignisflags etc., die sich auf den ersten Blick so nicht darlegen und erst in der Designphase zutage treten. Problematisch ist hier insbesondere der Umstand, daß die in den Datenblättern enthaltenen Spezifikationen zwar durchaus ausreichend sind für den Einsatz der entsprechenden Bausteine in einer Schaltung, jedoch für das Cloning meist nur unzureichende Informationen enthalten. Hier ist oftmals das Reverse-Engineering unvermeidlich, so daß man gestützt auf Datenblattinformationen, Erfahrungswerten und ggf. zusätzlicher Mewerte zunächst einmal ein Verhaltensmodell des zu klonenden Bausteins erstellen muß. Hierbei sollte man stets dem Modell den Vorzug geben, welches das Verhalten eines Bausteins mit minimalstem Aufwand beschreibt. Nicht zu empfehlen ist eine explizite Ausformulierung aller Bausteineinheiten, da dies meist zu aufgeblähten und nur relativ teuer zu implementierenden Verhaltensmodellen führt. Hilfreich ist hier stets die Fragestellung, wie sich das Verhalten unter Verwendung einfacher Logikfunktionen

realisieren läßt, da ein solch minimales Modell nicht nur geringste Anforderungen an das PLD stellt sondern auch eventuelle Seiteneffekte quasi von selbst liefert.

## 6 Cloning – wann sinnvoll?

Sicherlich ist Cloning nicht überall gleich interessant. Im großindustriellen Einsatz wird man diesen Weg sicher nicht beschreiten, da es hier sinnvoller ist, die Chipmasken bzw. Synthesemodelle dem ursprünglichen Hersteller abzukaufen, um auf dieser Basis eigene ASICs zu entwickeln.

Dieser Weg ist für kleine und mittlere Firmen indes selten gangbar, da die Lizenzkosten solcher Masken bzw. Modelle den zu erwartenden Gewinn bereits im Vorfeld mehr als auffressen können. Gerade bei Kleinserienfertigung, wie sie sich z.B. bei der Spezialisierung auf Nischenmärkte mit vergleichsweise niedrigen Umsatzzahlen ergibt, ist dieser Weg meist nicht gangbar. In einigen Fällen ist zudem die Aquirierung solcher Informationen nicht machbar, da die ehemaligen Hersteller in Konkurs gegangen bzw. aufgekauft worden sind und die neuen Besitzer der Lizenzrechte diese Informationen nicht herausgeben. In solchen Fällen bleibt nur der Weg der Modellierung anhand von Re-Engineering.

## 7 Quellen für Bausteinmodelle

Glücklicherweise ist dies in einigen Fällen nicht mehr nötig, da eine gewisse Anzahl von Modellen bereits im Internet frei verfügbar ist. Teilweise handelt es sich hier jedoch auch nur um reine Verhaltensmodelle, die nicht synthetisierbar sind, und manchmal sind die Modelle zwar synthetisierbar aber nicht vollständig. In jedem Falle sind diese Modelle jedoch eine gute Basis für eigene Verhaltensbeschreibungen. Eine weitere Quelle stellen auch Sourcecodes von Emulatoren dar, welche (im Idealfall) eine zyklusgenaue Bausteinbeschreibung enthalten. Hier bleibt einem zumindest die Modellerstellung erspart, zur Synthese müssen diese jedoch in eine geeignete HDL (Hardware Description Language) übersetzt werden. Zu beachten ist jedoch, daß Modelle in Programmiersprachen strikt sequentielle Ablaufstrukturen beinhalten, welche bei der Formulierung mithilfe von HDLs zu einem Gutteil parallelisiert werden können. Bei dieser Übersetzung ist unbedingt dafür Sorge zu tragen, daß das Modell nicht verändert wird, so daß sich weder funktionelles Verhalten als auch eventuell enthaltene Zustandsautomaten (beispielsweise zur Protokollsteuerung) nicht verändert werden.

## 8 Ausblick

Die immer stärkere Integration von Peripheriefunktionen in die Chipsätze moderner PC-Architekturen bedingt einen Wegfall von einfachen Peripheriebausteinen, wie sie

für den Embedded Controller-Markt nach wie vor interessant sein können. Doch auch hier geht man dazu über, derlei Funktionen mehr und mehr in die Controller selbst zu integrieren, was mittelfristig zu einem Wegfall solcher Peripheriebausteine führen wird. Betroffen hiervon sind vor allem kleinere Entwicklungsfirmen, für die solche Bausteine eine einfache und kostengünstige Möglichkeit zur Realisierung von Prototypen bzw. Kleinserien bieten.

Naheliegenderweise besonders betroffen von dieser Entwicklung sind Firmen, die sich auf die Reparatur älterer Geräte spezialisiert haben. Hier ist das Cloning von obsoleten Bausteinen mittelfristig die einzige Möglichkeit, einen Reparaturservice zu bieten. Dank des Preisverfalls im PLD-Sektor, der nicht nur die PLDs selbst sondern auch die hierfür nötige Entwicklungssoftware betrifft, ist hier das Cloning eine durchaus gangbare und rentable Methode.

## 9 Quellen

- **The VHDL Resource Page**  
*<http://www.eda.ei.tum.de/forschung/vhdl/index.html>*  
Sehr guter Einstiegspunkt zur Logiksynthese
- **The Hamburg VHDL Archive**  
*<http://tech-www.informatik.uni-hamburg.de/vhdl/vhdl.html>*  
Exzellente Quelle für VHDL-relevante Informationen

- **Free Model Foundry**  
*<http://vhdl.org/vi/fmf>*  
Sammlung frei verfügbarer VHDL- und Verilog-Modelle
- **RASSP Technology Base VHDL Modeling**  
*<http://www.erc.msstate.edu/mpl/rassp/html/overview.html>*  
Informationen zu VHDL sowie frei verfügbare Modelle
- **GNU Electronic Design Automation**  
*<http://www.geda.seul.org>*  
Freie EDA Software (in Entwicklung)
- **FreeHDL Server**  
*<http://www.freehdl.seul.org>*  
Freier VHDL-Simulator (in Entwicklung)