VHDL-Implementierung und FPGA-basierte Validierung eines digitalen PAL-Videosignal-Encoders

Lars Larsson

Universität Hamburg, Fachbereich Informatik / Arbeitsbereich TECH
Vogt-Kölln-Str. 30, D - 22527 Hamburg

Email: larsson@informatik.uni-hamburg.de
http://tech-www.informatik.uni-hamburg.de/~larsson/

Zusammenfassung

Der Beitrag beschreibt die VHDL-Implementierung eines rein digitalen PAL-Videosignal-Encoders. Die realisierte Architektur umfaßt neben einer Komponente zur Erzeugung der für die Bild-Synchronisation notwendigen Synchronisationssignale einen Generator zur Erzeugung des 4,43361875 MHz PAL-Farbträgers sowie einen QAM-Modulator und Videosignalmischer. Der PAL-Encoder liefert ein digitales FBAS-Signal mit einer Wortbreite von 8 Bit. Als analoge Schnittstelle kommt grundsätzlich jeder 8-Bit-Video-DAC in Frage. Es handelt sich bei dem digitalen PAL-Encoder um einen rein synchronen, vorderflankengetriggerten Entwurf. Die Taktfrequenz, die gleich der doppelten Abtastfrequenz ist, muß nicht zwingend ein ganzzahliges Vielfaches der Farbträgerfrequenz betragen, sondern ist grundsätzlich frei wählbar.

Bei steigender Integrationsdichte und sinkenden Signallaufzeiten von FPGAs werden in Zukunft DSP-Anwendungen mit FPGAs erschlossen werden können, die bisher mit programmierbaren Logikbausteinen nicht realisierbar sind. Der beschriebene Entwurf des digitalen PAL-Encoders ist ein Fallbeispiel für die Lösung einer Signalverarbeitungsaufgabe mit einem handelsüblichen FPGA-Baustein. Dabei wird gegenüber einer Software-Lösung eine um ein bis zwei Größenordnungen höhere Verarbeitungsgeschwindigkeit erreicht.

1 Einleitung

Auf Grund der Allgegenwärtigkeit analoger Farbfernsehgeräte kann die digitale Erzeugung eines PAL-Videosignals für Anwendungen, die graphische Ausgaben erfordern, wohl als eine Art Standardsignalverarbeitungsaufgabe verstanden werden. Die Verwendung von programmierbaren Logikbausteinen zur Lösung der vorliegenden DSP-Aufgabe weist verschiedene interessante Aspekte in Bezug auf Entwurfsmethodik und praktischer Randbedingungen auf. Rapid Prototyping mit FPGAs ist zunächst einmal eine kostengünstige Alternative zu kommerziellen Hardware-Simulationsbeschleunigern, die gerade für kleine Unternehmen geeignet ist. Im vorliegenden Fall dauert die funktionale VHDL-Simulation der Erzeugung einer Sequenz von acht Halbbildern etwa eine halbe Stunde. Mit einem in C geschriebenen PAL-Encoder-Programm erfordert die Erzeugung der Sequenz auf einem Celeron-333-MHz-PC unter Linux etwa 3,5 Sekunden. Der FPGA-Prototyp erzeugt die Bildsequenz von acht Halbbildern dagegen in nur 0,16 Sekunden – bei einer Taktfrequenz von nur 30 MHz. Zur Validierung eines digitalen PAL-Encoders genügt es nicht, kurze Signalverläufe – etwa einige Zeilen – zu oszillographieren. Erst die Darstellung von Bildsequenzen auf einem realen analogen Farbfernseher ermöglicht die Validierung und die Überprüfung der Spezifikation auf Robustheit. Kommerzielle digitale PAL-Encoder-Chips lassen meist nur einen Satz fester Taktfrequenzen zu. Der vorliegende

 Publiziert im Tagungsband der DSP Deutschland'99, München, 21.-23. September 1999, S. 315-324, Design & Elektronik, WEKA Fachzeitschriften-Verlag, Poing, Deutschland 1999.

FPGA-Prototyp ist dahingehend flexibler, da der Entwurf durch Änderung eines Parameters in der VHDL-Beschreibung und die Möglichkeit der Reprogrammierung des FPGAs grundsätzlich auf jede beliebige Taktfrequenz adaptierbar ist.

Da die Quadratur-Amplituden-Modulation (QAM) ein wesentlicher Bestandteil bei der Erzeugung eines PAL-Videosignals ist, entsteht ein digitaler QAM-Modulator bei der Implementierung eines PAL-Encoders als wiederverwendbares Nebenprodukt. Darüber hinaus erfordert die Erzeugung eines PAL-Videosignals die Erzeugung eines sinusförmigen Trägersignals mit schneller Phasenumtastung von hoher Frequenzgenauigkeit, jedoch mit einer Frequenz, die in einem besonders ungünstig (gewählten) Frequenzverhältnis zu üblichen Systemtaktfrequenzen liegt. Eine Komponente zur Erzeugung dieses sinusförmigen Trägersignals entsteht bei der Implementierung eines digitalen PAL-Encoders, so wie er im Rahmen des vorliegenden Beitrags konzipiert ist, ebenfalls als wiederverwendbares Nebenprodukt.

2 PAL-Videosignal

Die genaue zeitliche Struktur des PAL-Videosignals ist in Grundlagenliteratur zur Fernsehtechnik beschrieben [7, 9]. Weitere Detailinformationen der analogen Videosignalstandards (NTSC, PAL, SECAM) sind in [3] zu finden. Im folgenden werden die für den Entwurf einer PAL-Encoder-Architektur relevanten PAL-Videosignalkomponenten skizziert.

Das PAL-Videosignal setzt sich aus verschiedenen Signalkomponenten zusammen. Horizontalsynchronimpulse (H-SYNC) und Vertikalsynchronimpulse (V-SYNC) dienen der Synchronisation von Videobildquelle (Kamera) und Videobildziel (TV, Videorekorder). H-SYNC und V-SYNC werden zu Composite-Synchronimpulsen (C-SYNC) zusammengesetzt (s. Abbildung 2). Vor und nach dem V-SYNC werden zusätzliche H-SYNC-Impulse mit der halben H-SYNC-Impulslänge, aber doppelter H-SYNC-Frequenz eingefügt (Vortrabanten und Nachtrabanten), die der Signalverarbeitung analoger Fernsehgeräte dienen. Beim PAL-Verfahren werden 25 Bilder pro Sekunde übertragen. Zur Verringerung des Flimmerns, werden jedoch 50, vertikal um eine Zeile versetzte Halbbilder, d.h. Bilder mit der halben Zeilenzahl, übertragen. Durch das Zeilensprungverfahren wird die Verringerung des wahrnehmbaren Bildflimmerns erreicht, ohne die zur Bildübertragung erforderliche Videosignalbandbreite zu erhöhen. Bereiche außerhalb des sichtbaren Bildausschnitts (s. Abbildung 1) werden dunkel getastet. Die Erzeugung der Synchronisationssignale, Austastsignale und verschiedener Steu-

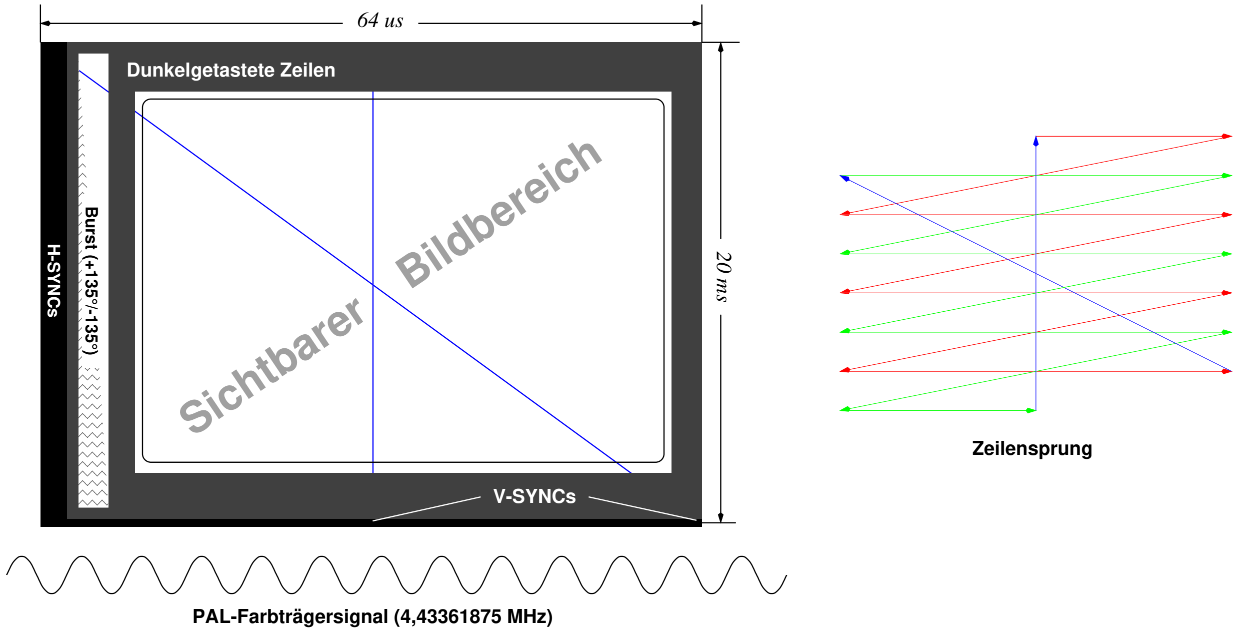

Abbildung 1: **BAS - Bild-Austast-Signale** *in schematischer Darstellung.*

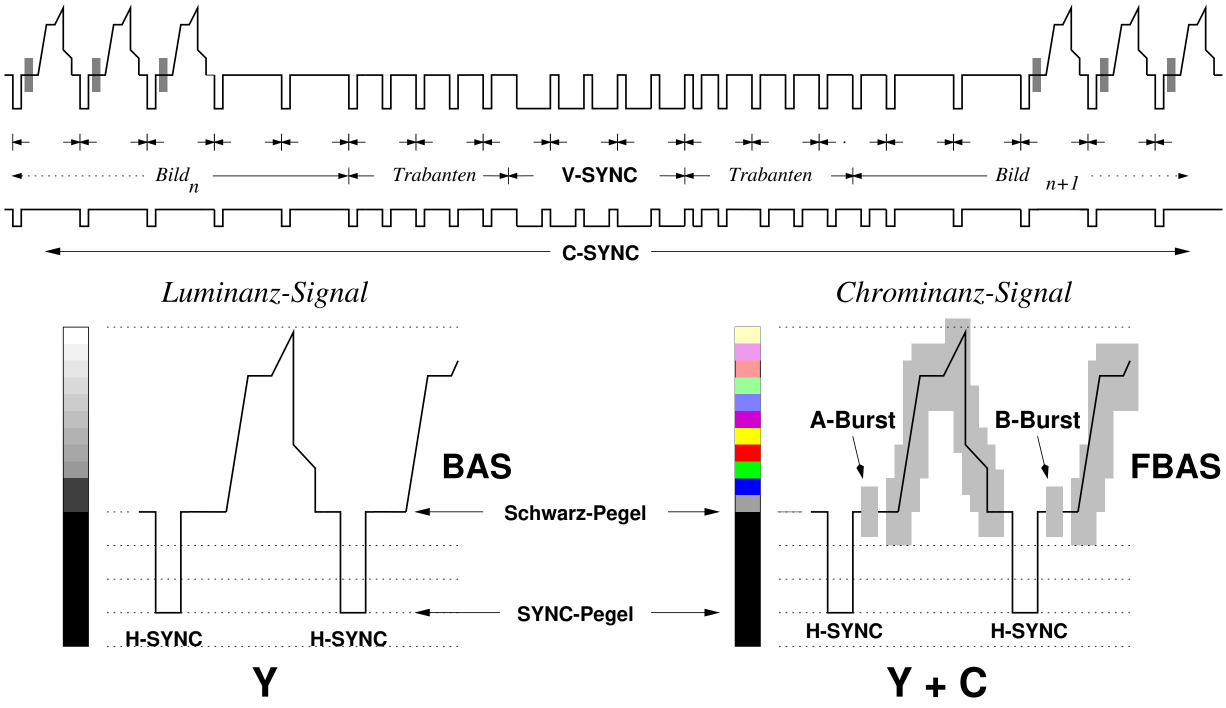

Abbildung 2: **FBAS - Farbe-Bild-Austast-Signal** *in schematischer Darstellung.*

ersignale (Farbträgerphase, Bildspeicheradressen, Bildnummer, interne Steuersignale, etc.) läßt sich leicht durch Zählerkaskaden realisieren.

Die Helligkeit von Bildpunkten wird durch die Amplitude der als Luminanz-Signal bezeichneten PAL-Videosignalkomponente repräsentiert. Das C-SYNC-Signal wechselt zwischen zwei Zuständen, nämlich zwischen Minimal-Pegel (SYNC-Pegel) des PAL-Videosignals und Schwarz-Pegel. Das Luminanz-Signal wird zu dem Schwarzpegel addiert. Das sich ergebende Signalgemisch zur Übertragung von Grauwertbildern wird als BAS-Signal (BAS $\approx$ **B**ild-**A**ustast-**S**ynchronsignal [7]) bezeichnet. Die Erweiterung des BAS-Signals durch zusätzliche Signalkomponenten für die Farb-Übertragung wird als FBAS-Signal (FBAS $\approx$ Farb-BAS [7]) bezeichnet. BAS-Signal und FBAS-Signal sind in Abbildung 2 skizziert.

Die Farbübertragung des PAL-Standards erfolgt nicht direkt auf Basis des RGB-Farbraums sondern auf Basis des YUV-Farbraums. Dazu werden ergänzend zur Bildpunkthelligkeit (*Luminanz*) y zwei Farbdifferenzsignale u und v übertragen. RGB-Farbraum und YUV-Farbraum sind durch lineare Transformationen ineinander überführbar [3].

$$\begin{array}{rcl} y &=& 0,30 \cdot r + 0,59 \cdot g + 0,11 \cdot b \\ u &=& b - y \\ v &=& r - y \end{array} \qquad \begin{array}{rclll} r &=& y & & +1,14 \cdot v \\ g &=& y & -0,39 \cdot u & -0,58 \cdot v \\ b &=& y & +2,04 \cdot u & \end{array} \tag{1}$$

Die beiden Farbdifferenzsignale u und v werden mittels Quadratur-Amplituden-Modulation [7] übertragen. Zur Vermeidung bzw. Reduzierung von Über- und Untersteuerung werden die Farbdifferenzsignale zuvor gemäß $u := u \cdot 0,49$ und $v := v \cdot 0,88$ reduziert. Die Quadratur-Amplituden-Modulation

$$c(t) = u(t) \cdot \cos(2\pi \cdot f_{PAL} \cdot t) \pm v(t) \sin(2\pi \cdot f_{PAL} \cdot t) \tag{2}$$

liefert das *Chrominanz*-Signal c. Dabei ist $f_{PAL} = 4,43361875\ MHz$ die Frequenz des PAL-Farbträgersignals. Das Vorzeichen des Farbdifferenzsignals v wird beim PAL-Verfahren von Zeile zu Zeile gewechselt, wodurch das PAL-Verfahren weniger anfällig gegenüber phasenfehlerbedingten Farbfehlern als das verwandte NTSC-Verfahren ist [7, 9]. Zur Rekonstruktion der Farbdifferenzsignale u und v in einem PAL-Decoder ist die genaue Kenntnis der Phase und des Vorzeichens von v des Farbträgersignals im PAL-Encoder erforderlich. Im PAL-Encoder wird dann das Chrominanz-Signal $c(t)$ mit $\cos(\omega \cdot t)$ und mit $\sin(\omega \cdot t)$, wobei $\omega = 2\pi \cdot f_{PAL}$, multipliziert.

$$
\begin{aligned}
d(t) &= c(t)\cdot\cos(\omega\cdot t) &= \tfrac{1}{2}u(t)+\tfrac{1}{2}u(t)\cdot\cos(2\omega\cdot t)+\tfrac{1}{2}v(t)\cdot\sin(2\omega\cdot t) \\
e(t) &= c(t)\cdot\sin(\omega\cdot t) &= \tfrac{1}{2}v(t)+\tfrac{1}{2}u(t)\cdot\sin(2\omega\cdot t)-\tfrac{1}{2}v(t)\cdot\cos(2\omega\cdot t)
\end{aligned}
\qquad (3)
$$

Durch Tiefpaßfilterung werden die Terme in (3) mit $\cos(2\omega\cdot t)$ und $\sin(2\omega\cdot t)$ im PAL-Decoder unterdrückt, so daß die Terme $\frac{1}{2}u(t-T_G)$ und $\frac{1}{2}v(t-T_G)$ von (3) übrig bleiben. Das Luminanz-Signal y wird um die Gruppenlaufzeit T_G der Tiefpaßfilter verzögert. Neben der Kenntnis der Phase des PAL-Farbträgers ist die Kenntnis des zeilenweise wechselnden Vorzeichens ($\pm$) des Farbdifferenzsignals v in (2) notwendig. Dazu wird beim PAL-Verfahren die Phase des Burst-Signals (s. Abbildung 3) von Zeile zu Zeile zwischen $+135^o$ (sog. A-Burst) und -135^o (sog. B-Burst) hin und her geschaltet. Die Phase des Farbträgers wird für die QAM-Modulation (2) jedoch nicht hin und her geschaltet. Der Farbträgersignalgenerator muß daher ein sinusförmiges Farbträgersignal $f(t)$ mit vier verschiedenen Phasen zur Verfügung stellen. Je nach Abtastzeitintervall (A-Burst, B-Burst, Bildbereich ($+u$, $\pm v$)) wird ein sinusförmiges Farbträgersignal

$$
f(t) = \begin{cases}
\cos(\omega t - 0^o) & : \text{u-Träger } (u(t)\cdot\cos(\omega t)) \\
\cos(\omega t - (+90^o)) & : \text{v-Träger } (v(t)\cdot\sin(\omega t)) \\
\cos(\omega t - (+135^o)) & : \text{A-Burst-Träger} \\
\cos(\omega t - (-135^o)) & : \text{B-Burst-Träger}
\end{cases}
\qquad (4)
$$

ausgewählt und nachgeschalteten Signalverarbeitungseinheiten bereit gestellt. Das Farbträgersignal muß unabhängig vom Zeitraster der Synchronisationssignalen zur Verfügung gestellt werden. Das Farbträgersignal (4) wird mit der für die unterschiedlichen Zeitintervallen (A-Burst, B-Burst, sichtbarer Bildbereich) zugeordneten Phase ausgewählt und dem BAS-Signal hinzugefügt, wie in Abbildung 3 skizziert.

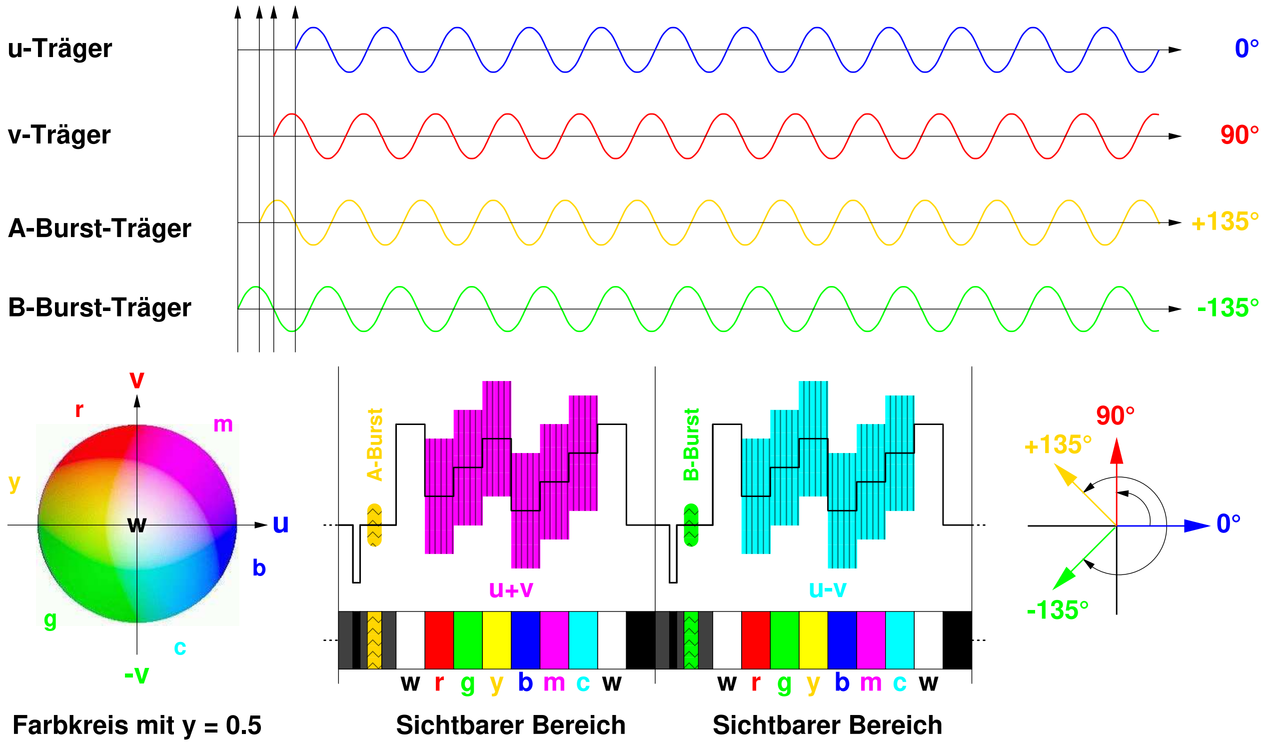

Abbildung 3: **PAL-Videosignal - Phasenbeziehungen**

3 PAL-Encoder

Die Aufgabe eines PAL-Encoders ist die Konvertierung von Farbbildinformation in ein Signalgemisch, das zur Darstellung dieser Farbbildinformation auf analogen PAL-Farbfernsehern geeignet ist. Ein PAL-Encoder konvertiert einen Bilddatenstrom

$$(r(t), g(t), b(t)) \longrightarrow (y(t), u(t), v(t)) \longrightarrow \text{fbas}(t). \tag{5}$$

Die im vorliegenden Beitrag beschriebene Implementierung eines digitalen PAL-Encoders konvertiert einen YUV-Bilddatenstrom in einen digitalen FBAS-Bilddatenstrom mit einer Datenwortbreite von $8\ Bit$. Ein PAL-Encoder ist immer dann erforderlich, wenn Bildinformationen auf *analogen* PAL-Videogeräten dargestellt oder weiter verarbeitet werden sollen.

3.1 Komponenten

In Abbildung 4 ist die als FPGA-Prototyp implementierte Architektur skizziert. Der PAL-Encoder-Kern setzt sich aus vier Komponenten zusammen, die jeweils in einer VHDL-Entity beschrieben sind. Der PAL-Encoder-FPGA-Prototyp verfügt über drei Betriebsmodi, die durch einen Schalter (SW) selektiert werden können. Im ersten Betriebsmodus kann eine Farbe aus dem YUV-Farbraum durch DIP-Schalter ausgewählt werden (s. Abbildung 6). Dabei kann der gewünschte Punkt im YUV-Farbraum mit einer Genauigkeit von insgesamt $16\ Bit$ spezifiziert werden. Durch die endliche Zahlendarstellungsgenauigkeit ist nur eine Teilmenge der Farben des YUV-Farbraums darstellbar. Das gilt insbesondere für Farben mit großen Beträgen von u und v bei großem und kleinem y. Mögliche interne Zahlenbereichsüberschreitungen werden jedoch begrenzt. Durch einen zusammengesetzten Bit-Vektor der Form `y(7 downto 0) & u(3 downto 0) & v(3 downto 0)` erfolgt die Farbauswahl. Dabei ist `y` vom Typ `unsigned`, `u` und `v` sind vom Typ `signed` (ieee.std_logic_1164, ieee.std_logic_signed). Im zweiten Betriebsmodus wird y von einem Bildzähler geliefert, u und v werden aus höherwertigen Bits von Zeilen- und Spaltenadressen abgeleitet. So wird der gesamte adressierbare YUV-Farbraum durchlaufen. Im dritten Betriebsmodus wird der Bildzähler angehalten, so daß eine Ebene im YUV-Farbraum permanent dargestellt wird.

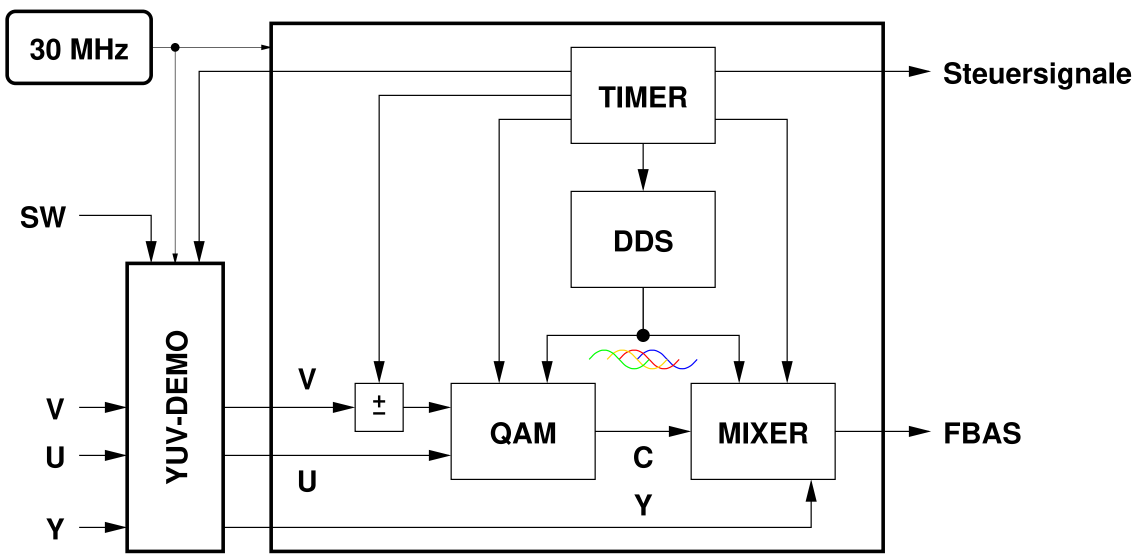

Abbildung 4: **PAL-Encoder-Architektur**

3.1.1 Timer

Der TIMER erzeugt alle für die Erzeugung eines FBAS-Videosignals notwendigen Signalkomponenten sowie interne Steuersignale für den PAL-Encoder. Außerdem werden Adressen zur Ansteuerung von externen SRAMs erzeugt, die aber bei dem FPGA-Prototypen nicht herausgeführt sind.

3.1.2 Farbträger-Generator

Die Komponente DDS ($\approx$ **D**irekte **D**igital **S**ynthese) erzeugt das PAL-Farbträgersignal (4) durch fortschreitende Akkumulation einer Phasenkonstante $\Delta\varphi$ mit jedem Abtastschritt (vgl. *Quadrature Subcarrier Generation* mit *Ratio Counter* [3]). Beim vorliegenden Entwurf wird jedoch anstelle einer in einem ROM abgelegten Sinus-Tabelle diese durch ein Schaltnetz (synthetisierbare VHDL-Beschreibung) repräsentiert. Dadurch kann bei der Schaltungssynthese auf (ROM-)Generatoren verzichtet werden, wodurch die Wiederverwendbarkeit etwas vereinfacht werden kann [6]. Ein Farbträgersignal (4) läßt sich allgemein durch den Ausdruck

$$f(t,\Phi) \;=\; \sin(\underbrace{2\pi \cdot f_{PAL} \cdot t}_{\varphi(t)} + \Phi) \tag{6}$$

beschreiben. Durch geeignete Wahl der Phasenverschiebungskonstante Φ kann das Farbträgersignal (4) mit einer der vier benötigten Phasenlagen erzeugt werden. Zur Vereinfachung wird im folgenden $\Phi := 0$ gesetzt. Das Funktionsargument in (6) ist ein Phasenwinkel $\varphi(t)$ zum Zeitpunkt t. Durch die zeitdiskrete Abtastung mit einer Abtastfrequenz $f_s = 1/\Delta t_s$ ergibt sich eine Folge von Phasenwinkeln $\varphi(t) = \{\varphi(t_0), \varphi(t_1), \varphi(t_2), \ldots, \varphi(t_i)\}$ zu diskreten Zeitpunkten $t = \{t_0, t_1, t_2, \ldots, t_i\}$. Zum i-ten Abtastzeitpunkt ist daher $t_i = i \cdot \Delta t_s$. Für den Phasenwinkel φ_i beim i-ten Abtastzeitpunkt gilt also

$$\varphi_i \;=\; i \cdot \Delta\varphi = \sum_{j=0}^{i-1} \Delta\varphi, \tag{7}$$

wobei $\Delta\varphi = 2\pi \cdot f_{PAL} \cdot \Delta t_s = 2\pi \cdot \frac{f_{PAL}}{f_s}$. Damit errechnet sich das Trägersignal f_i beim i-ten Abtastzeitpunkt t_i zu $f_i = \sin(\varphi_i)$ bzw. $f_i = \sin(\varphi_i + \Phi)$. Die benötigte Sinus-Tabelle `sin[p]` enthält *genau eine* Sinus-Periode mit `N` Tabelleneinträgen und wird gemäß

$$\texttt{sin[p]} := \sin(2\pi \cdot \texttt{p}/\texttt{N}) \qquad \text{mit} \qquad \texttt{p} = 0, 1, 2, \ldots, N-1 \tag{8}$$

berechnet. Es ist zwar nicht zwingend notwendig, aber zweckmäßig, für `N` eine Zweierpotenz zu wählen, so daß für natürliche Zahlen `n`, `N` gilt $2^{\texttt{n}} = \texttt{N}$. Die Darstellungsgenauigkeit der Tabelleneinträge in der Sinus-Tabelle legt den durch Quantisierungsrauschen bestimmten Signal-Rausch-Abstand eines erzeugten Sinus-Signals fest. Ein Sinus-Signal, das mit einer Wortbreite von n Bits (bei linearer, ganzzahliger Darstellung) quantisiert ist, weist einen Signal-Rausch-Abstand von $\approx 1{,}8$ [dB] $+ \, n \cdot 6$ [dB] auf [12]. Dazu ist jedoch die Zahl der Tabelleneinträge derart zu wählen, daß der Differenzbetrag aufeinander folgender Tabelleneinträgen $|\texttt{sin}[\texttt{i}] - \texttt{sin}[\texttt{i}+1]| \leq 1.0$ ist, da $|\frac{d}{dx}\sin(x)| \leq 1.0$ ist. Diese Bedingung ist erfüllt, wenn die Zahl der Tabelleneinträge $\texttt{N} > 2^n \cdot \pi$ ist, da das Seitenverhältnis einer vollen Periode ($2\pi \approx \texttt{N}$) zur Sinus-Auslenkung ($\pm 1 \approx 2^{\texttt{n}}$) gerade $2\pi/2$ beträgt. Der Phasenwinkel φ_i zum i-ten Abtastzeitpunkt wird mit Hilfe des Phasenakkumulators `A` in vorzeichenloser Zahlendarstellung gemäß

$$\texttt{A}_i := (\texttt{A}_{i-1} + \texttt{C}) \bmod 2^{\mathsf{a}} \quad \text{wobei} \quad \texttt{C} = 2^{\mathsf{a}} \cdot \frac{f_{PAL}}{f_s} \qquad \text{mit} \qquad \texttt{A}, \texttt{C} \in [0, \ldots, 2^{\mathsf{a}} - 1] \tag{9}$$

berechnet. Dabei ist a die Wortbreite des Phasenakkumulators. Die erreichbare Frequenzgenauigkeit des Farbträgersignals, d.h. die Genauigkeit, mit der das Frequenzverhältnis f_{PAL}/f_s darstellbar ist, hängt von der Wortbreite der Akkumulationskonstanten `C` ab. Durch Veränderung der Konstanten `C` läßt sich der digitale PAL-Encoder einfach auf unterschiedliche Abtastfrequenzen f_s adaptieren. Der Sinus-Wert zum i-ten Abtastzeitpunkt ist

$$\mathrm{sin}_i = \mathtt{sin[\ A}_i\mathtt{((a-1)\ downto\ (a-n))\ +\ }\Phi\ \mathtt{]} \tag{10}$$

Da die Wortbreite des Adressvektors der Sinus-Tabelle deutlich geringer als die Wortbreite des Phasen-Akkumulators `A` ist, werden zur Adressierung der Sinus-Tabelle (10) nur die höchstwertigen Bits `((a−1) downto (a−n))` verwendet.

3.1.3 QAM-Modulator

Der QAM-Modulator moduliert die Farbdifferenzsignale $u(t)$ und $v(t)$ gemäß (2) und erzeugt so das Chrominanz-Signal $c(t)$. Dabei wird die Berechnung (2) bei jedem zweiten Takt ausgewertet. Daher wird der digitale PAL-Encoder mit der doppelten Frequenz der für das FBAS-Signal gewünschten Frequenz des FBAS-Signals betrieben. Der QAM-Modulator ist als MAC-Einheit (**M***ultiply-and-***AC***cumulate*) implementiert. Dadurch wird zur Berechnung von (2) nur *ein* Multiplizierer benötigt, der wegen der hohen erforderlichen Verarbeitungsgeschwindigkeit als Parallel-Multiplizierer (Schaltnetz) realisiert werden muß. Bei der ersten (τ_0) von zwei ($\ldots, \tau_0, \tau_1, \ldots$) aufeinander folgenden Taktflanken wird die MAC-Einheit mit $u(t) \cdot cos(\omega t)$ initialisiert. Bei der zweiten Taktflanke (τ_1) der beiden aufeinander folgenden Taktflanken ($\ldots, \tau_0, \tau_1, \ldots$) wird $v(t) \cdot sin(\omega t)$ zu dem im Akkumulator-Register `A` gespeicherten Wert hinzu addiert. Die Weitergabe des im Akkumulator gespeicherten Wertes an den Videomischer erfolgt mit jedem zweiten Takt, so daß gilt

$$\begin{array}{rcl} A_{\tau_0} & := & u(t) \cdot \cos(\omega \cdot \tau_0) \\ A_{\tau_1} & := & A_{\tau_0} \pm v(t) \cdot \sin(\omega \cdot \tau_0). \end{array} \tag{11}$$

Dabei sind a_{τ_0} und a_{τ_1} die Werte des Akkumulator-Registers zweier aufeinander folgender Takte. Da die Weitergabe des Akkumulator-Inhalts nur mit jeder zweiten Taktflanke erfolgt, ist $c_{\tau_0} = c_{\tau_1} = A_{\tau_1}$.

3.1.4 Videomischer

Aufgabe des Videomischers MIXER ist es, ein vorzeichenloses FBAS-Signal durch Addition von konstantem Schwarzpegel, Luminanz-Signal y und Chrominanz-Signal c zu berechnen. Darüber hinaus kommt dem MIXER die Aufgabe der Selektion der Burst-Signale zu. Dazu werden geeignete Steuersignale vom TIMER generiert. Außerdem überwacht der MIXER Zahlenbereichsüberschreitungen und begrenzt diese gegebenenfalls auf den gültigen FBAS-Zahlenbereich ($0 \ldots 255$). Dabei ist entscheidend, daß die Addition des Chrominanz-Signals zum Luminanz-Signal *vorzeichenbehaftet* erfolgt, *bevor* das FBAS-Signal in ein vorzeichenloses Signal konvertiert wird. Wird das Chrominanz-Signal vor der Addition in ein vorzeichenloses Signal überführt, so hat die Addition nicht die erforderliche Semantik. Vom TIMER wird die Auswahl der Signalkomponenten durch Steuersignale csync, black, burst bestimmt, von denen immer nur ein Signal aktiv ist, mit Ausnahme von black. Die Signale csync und burst haben jedoch gegenüber dem Signal black Priorität.

$$\mathrm{fbas}(t) \;=\; \begin{cases} \textsf{SYNC-Pegel} & \text{wenn} \quad \textsf{csync} = aktiv \\ \textsf{Schwarz-Pegel} & \text{wenn} \quad \textsf{black} = aktiv \\ \textsf{Schwarz-Pegel} + \frac{1}{8} \cdot f(t, \Phi) & \text{wenn} \quad \textsf{burst} = aktiv \\ \textsf{Schwarz-Pegel} + y(t) + c(t) & \text{sonst.} \end{cases} \tag{12}$$

Für den Fall, daß $\mathrm{fbas}(t)$ kleiner als null wird, wird $\mathrm{fbas}(t)$ auf null gesetzt. Ergibt sich für $\mathrm{fbas}(t)$ ein Wert größer als $2^8 - 1$, so wird $\mathrm{fbas}(t)$ auf $2^8 - 1$ gesetzt. Dazu erfolgen die Berechnungen (12) mit hinreichend großen Wortbreiten zur Vermeidung von Zahlenbereichsüberschreitungen, da für die VHDL-Verhaltensbeschreibung der Operator + direkt verwendet wird, um die Lesbarkeit der VHDL-Beschreibung zu erhalten. Die Handhabung von Carry-Signalen zur Überlaufbehandlung wurde in diesem Zusammenhang gar nicht erst in Betracht gezogen. Das digitale FBAS-Signal wird durch einen 8-Bit-Video-DAC vom Typ Philips TDA8702 [10] in ein analoges FBAS-Videosignal konvertiert.

4 Design-Ablauf

Grundlage der vorliegenden VHDL-Implementierung des digitalen PAL-Encoders ist eine in der Sprache C geschriebene ausführbare Spezifikation eines digitalen PAL-Encoders. Dieses Programm wurde vom Autor des vorliegenden Beitrags im Rahmen betreuter Studien- und Diplomarbeiten [4, 5] geschrieben. Die ausführbare Spezifikation des digitalen PAL-Encoders diente primär der Validierung des Signalverarbeitungsschemas eines digitalen PAL-Encoders. Das C-Programm wurde mit der bidirektionalen Videosignalschnittstelle *VidTrans* [8] in realer Systemumgebung unter Echtzeitbedingung validiert. Die Validierung der ausführbaren Spezifikation des digitalen PAL-Encoders durch Darstellung digital erzeugter PAL-Videosignale mit der Videosignalschnittstelle *VidTrans* hat sich bei der vorliegenden Signalverarbeitungsaufgabe als unverzichtbares Hilfsmittel erwiesen. Darüber hinaus konnte so auch die hohe, durch einen digitalen PAL-Encoder grundsätzlich erreichbare Bildqualität *vor* der Hardware-Implementierung evaluiert werden. In Abbildung 5 sind synthetische Farbbilder, mit der C-Implementierung des digitalen PAL-Encoders erzeugte Videosignalausschnitte und die mit einer C-Implementierung eines digitalen PAL-Decoders rück-extrahierten Farbbilder dargestellt. Die Unterschiede zwischen den Farbbildern sind derart gering, daß sie bei Ausgabe über handelsübliche Farbtintendrucker oder Farblaserdrucker nicht wahrnehmbar sind. Quantisierungsartefakte lassen sich erst bei Darstellung auf analogen Farbfernsehgeräten beobachten.

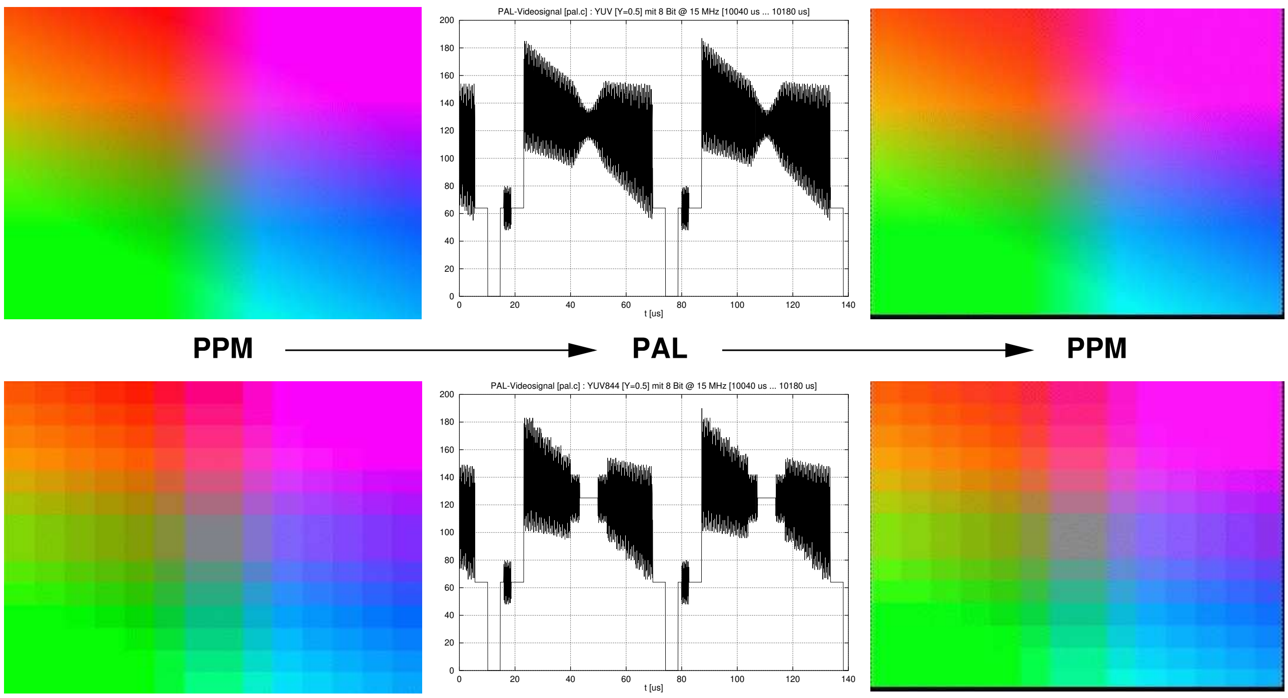

Abbildung 5: **Software-PAL-Encoder und Software-PAL-Decoder** *zur Evaluierung der Videosignalverarbeitungsaufgabe. Im oberen Teil der Abbildung wurde eine Ebene des YUV-Farbraums mit einem kontinuierlichen Farbverlauf in ein PAL-Videosignal konvertiert. Im unteren Teil der Abbildung wurde die gleiche Ebene in ein PAL-Videosignal konvertiert, jedoch mit der Darstellungsgenauigkeit von* u *und* v*, wie sie bei der VHDL-Implementierung des PAL-Encoders verwendet wurde.*

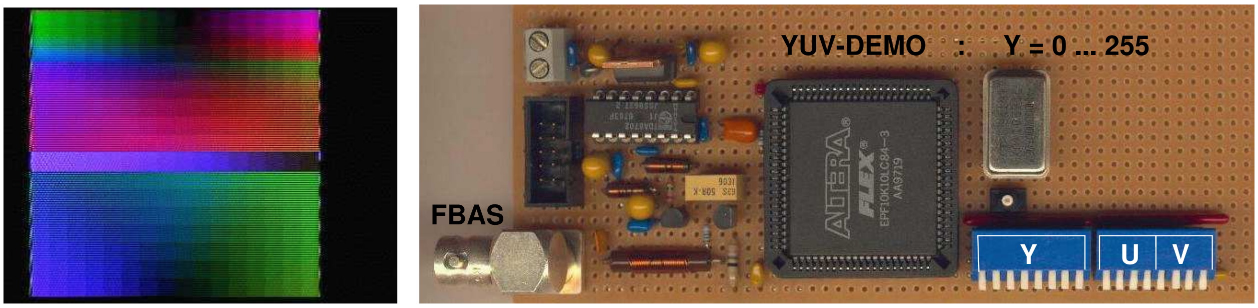

Abbildung 6: **PAL-Video-Bild und PAL-Encoder-FPGA-Prototyp-Board.** *Auf der linken Seite der Abbildung ist ein PAL-Videobild als Ergebnis einer funktionalen VHDL-Simulation dargestellt, wie es der Prototyp des FPGA-PAL-Encoders liefert. Das PAL-Videobild wurde mit einem Software-PAL-Decoder aus dem Videosignaldatenstrom der VHDL-Simulation extrahiert.*

4.1 Top-Down-Architektur-Konzept – Bottom-Up-Implementierung

Die Konzipierung der Architektur (Abbildung 4) des digitalen PAL-Encoders erfolgte Top-Down. Der Entwurf der Komponenten erfolgte dagegen Bottom-Up in der Reihenfolge: TIMER, DDS, QAM, MIXER. Abschließend wurde die Komponente YUV-DEMO entworfen. Für den vorliegenden PAL-Encoder wurde die benötigte Sinus-Tabelle mit einem C-Programm als VHDL-Beschreibung (ein Prozeß mit `case` Statement in einer VHDL-Entity) generiert, die das Verhalten eines Schaltnetzes impliziert [6, 2] und von der Logiksynthese nach Möglichkeit als einstufiges Schaltnetz realisiert wird oder zumindest in ein Schaltnetz mit wenig Stufen umgesetzt wird.

Für jede Komponente wurde nach der Fertigstellung eine eigene Testumgebung implementiert, um die Komponenten vor der Instantiierung intensiv durch Verhaltenssimulation zu validieren. Jede Komponente wurde nach der Validierung synthetisiert, um die Synthetisierbarkeit der Komponenten und später des gesamten PAL-Encoders zu gewährleisten. Abschließend wurde die VHDL-Beschreibung des PAL-Encoders zusammen mit einer geeigneten Testumgebung erstellt. Für die Testumgebung des PAL-Encoders wurde ein vereinfachtes Verhaltensmodell des verwendeten 8-Bit-Video-DACs (TDA8702 [10]) erstellt, welches die Abtastwerte des digitalen FBAS-Datenstroms in eine Textdatei schreibt. Diese Textdatei dient als Basis für die graphische Signaldarstellung (wie in Abbildung 5) oder Videobildextraktion (wie in Abbildung 6) des erzeugten PAL-Videosignals und zur Darstellung des erzeugten PAL-Videosignals auf einem analogen Farbfernsehgerät mit Hilfe von *VidTrans* [8].

4.2 FPGA-Prototyp-Board

Zur Validierung des VHDL-Modells in der Systemumgebung wurde eine Prototyp-Platine (s. Abbildung 6) innerhalb eines Arbeitstages in Fädeltechnik aufgebaut. Der zügige Aufbau des Prototyp-Boards wurde durch die Verwendung nur eines einzigen FPGAs, welches hinreichend viele Gatter-Äquivalente zur Verfügung stellt, ermöglicht. Neben dem FPGA (Altera EPF10K10LC84-3, 10 K Gatteräquivalente [1]) befindet sich auf dem Prototyp-Board noch ein 30-MHz-Quarzoszillator, einige Schalter und Buchsen, sowie der 8-Bit-Video-DAC (Philips TDA8702 [10]). Die Beschaltung des Video-DACs mit einigen analogen Komponenten wurde im wesentlichen dem Video-DAC-Datenblatt [10] entnommen.

5 Schlußfolgerungen

Eine synthetisierbare VHDL-Verhaltensbeschreibung eines digitalen PAL-Encoders, die sogar für die Implementierung als FPGA-Prototyp geeignet ist, ist dann für digitale System-on-a-Chip-Entwürfe (SoC) geeignet, wenn beim Entwurf Leitlinien für SoC-Designs [6] berücksichtigt werden. So wurde beim vorliegenden Entwurf auf die Ausnutzung von herstellerspezifischen Merkmalen des zum

Prototyping verwendeten FPGAs verzichtet. Es wurde ein rein synchroner Entwurf mit Einflanken-Triggerung realisiert, und es wurden nur Moore-Automaten verwendet. Dadurch steht das Verhalten der funktionalen Simulation mit dem Verhalten des synthetisierten FPGAs, das im System unter Echtzeitbedingungen validiert werden konnte, im Einklang.

Standardzellen- und Full-Custom-Entwürfe [13] bieten Möglichkeiten, die beim Prototyping mit FPGAs nicht zwingend zur Verfügung stehen. So ist etwa die Verwendung von größeren On-Chip-Speichern bei FPGAs im allgemeinen nicht effizient möglich. Zwar unterstützen FPGAs der Flex10K-Familie [1] bis zu einem gewissen Grade die Verwendung von On-Chip-RAM, jedoch nur als herstellerspezifische Makroblocks.

Ein Vorteil bei der Verwendung eines durch FPGA-Prototyping validierten VHDL-Modells eines PAL-Encoders anstelle eines handelsüblichen, digitalen PAL-Encoders, ist die Möglichkeit der Wiederverwendbarkeit für SoC-Designs. Darüber hinaus bietet es Unabhängigkeit von Herstellern digitaler PAL-Encoder, was bei Verwendung des betreffenden VHDL-Modells anstelle eines Chips unter Umständen ein Redesign bei Einstellung der Produktion eines verwendeten Chips vermeiden hilft [11]. Die Verwendung von FPGAs für Prototyping, durch Einbau von FPGAs in ein Zielsystem, bietet die Möglichkeit, einen digitalen Entwurf in realer Systemumgebung unter Echtzeitbedinungen zu validieren, wie das vorliegende Fallbeispiel zeigt.

Literatur

[1] Altera Corporation (1998). *Altera – FLEX 10K – Embedded Programmable Logic Family – Data Sheet Ver. 3.13*. San Jose, California, USA.

[2] J. Bhasker (1996). *A VHDL Synthesis Primer*. Star Galaxy Publishing, Allentown, Pennsylvania.

[3] K. Jack (1996). *Video Demystified: A Handbook for the Digital Engineer*. HighText, San Diego, California, USA, 2. Auflage.

[4] S. Jürgens (1996). Simulation rein digitaler Farbseparation (YUV, RGB) aus einem Videosignal im PAL-Standard zur Abschätzung des Implementationsaufwands. Studienarbeit, Universität Hamburg, Fachbereich Informatik.

[5] S. Jürgens (1999). Systemnahe Simulation rein digitaler Verarbeitung analoger Farbvidoesignaldaten (PAL). Diplomarbeit, Universität Hamburg, Fachbereich Informatik.

[6] M. Keating, P. Bricaud (1998). *Reuse Methodology Manual for System-On-A-Chip Designs*. Kluwer Academic Publishers.

[7] L. Kirsch (1993). *Fernsehtechnik*. Vieweg, Braunschweig / Wiesbaden.

[8] L. Larsson (1996). An EPLD Based Transient Recorder for Simulation of Video Signal Processing Devices in a VHDL Environment Close to System Level Conditions. In *Proc. of Sixth Int. Workshop on Field Programmable Logic and Applications, FPL'96, Darmstadt, Germany, 23-25 September 1996*, R. W. Hartenstein, M. Glesner, Eds., *Lecture Notes in Computer Science*, Band 1142, S. 371–375,. Springer-Verlag, Heidelberg.

[9] R. Mäusel (1995). *Fernsehtechnik*. Hüthing Buch Verlag, Heidelberg, 2. Auflage.

[10] Philips Semiconductor, *TDA8702 Data Sheet 1997*, Philips Semiconductor.

[11] R. C. Stogdill (1999). *Dealing with Obsoleta Parts*, in IEEE Design & Test of Computers, Ausgabe April-June 1999, S. 17-25, IEEE Circuits and Systems Society, Los Alamitos, California.

[12] U. Tietze, C. Schenk (1993). *Halbleiter-Schaltungstechnik*. Springer-Verlag, Heidelberg, 10. Auflage.

[13] N. H. E. Weste, K. Eshraghian (1993). *Principles of CMOS VLSI Design – A Systems Perspective*. Addison-Wesley, 2. Auflage.