

fbas_enc: Farbvideosignalerzeugung mit einem CPLD

V0.31 (c) 2007 Jörg Wolfram

1 Lizenz

Das Programm unterliegt der LGPL (GNU Lesser General Public Licence) Version 2 oder höher, jede Nutzung der Software/Informationen nonkonform zur LGPL oder ausserhalb des Geltungsbereiches der LGPL ist untersagt! Die Veröffentlichung dieses Programms erfolgt in der Hoffnung, daß es Ihnen von Nutzen sein wird, aber OHNE IRGENDNE GARANTIE, auch ohne die implizite Garantie der MARKTREIFE oder der VERWENDBARKEIT FÜR EINEN BESTIMMTEN ZWECK.

2 Geschichte und Features

Nach einigen Projekten zur Farbausgabe mit AVR-Mikrocontrollern war auch die Frage entstanden, ob man auf einfachem Wege ein FBAS-Signal erzeugen könnte. Zum Beispiel, um ein moduliertes HF-Signal in eine Antennenanlage einspeisen zu können. Da ein Mikrocontroller dafür aufgrund der notwendigen Verarbeitungsgeschwindigkeit eher ungeeignet ist, fiel die Entscheidung zugunsten eines CPLD. Natürlich gibt es für solche Dinge auch Spezial-IC's, aber diesen Weg wollte ich nicht wählen. Herausgekommen ist letztendlich ein sehr simples Design, welches auch in einen XC9536 von Xilinx passt. Der Encoder kann auch in andere Designs integriert werden, vorausgesetzt, die Bestimmungen der LGPL werden eingehalten.

- Gewinnung der Farbträgerfrequenz aus dem Systemtakt von 16/20/32/40 MHz
- Eingänge: hsync, vsync und rgb
- 8 Grundfarben / 8 Graustufen wählbar
- Gleiche Hardware kann für PAL und NTSC verwendet werden, auch umschaltbar
- einfache Einbindung als Komponente in andere Designs

3 Ein bisschen Theorie

Zum Beschreibung von Farbvideosignalen gibt es viele Adressen im Internet und so will ich nur kurz das Wichtigste anhand des PAL-Systems anreissen.

Horizontal- und Vertikaltiming sind dem schwarzweissen BAS-Signal weitestgehend identisch, was hinzukommt, ist der Farbträger. Damit dieser auf SW-Geräten möglichst nicht sichtbar ist, und um Moires zu vermeiden wurde eine „krumme“ Frequenz gewählt. Und zwar die Horizontalfrequenz von 15625 Hertz * 283,75 plus die halbe Vertikalfrequenz (25 Hertz). Damit kommt man auf 4433618,75 Hertz. Dafür könnte man nun einen Quarzoszillator benutzen oder gewinnt die Frequenz aus dem Systemtakt. Dies geschieht mittels DDS (Direct Digital Synthesis) wobei ein Fehler von eben den obengenannten 25 Hz in Kauf genommen werden muß, oder man benötigt sehr breite Zähler zur Signalerzeugung. Bei einer Taktfrequenz von 16 MHz braucht man so einen Zähler (Akkumulator) mit 12 Bit Breite. Bei jedem Takt wird zum Zähler 1135 (283,75 * 4) dazuaddiert, und schon hat man die (ungefähre) Farbträgerfrequenz. Bei 20 MHz wären es dann theoretisch 908, es würde auch ein 10 Bit breiter Zähler mit Addition von 227 genügen. Das Ganze funktioniert dann so, dass der Zähler zu jedem Abtastzeitpunkt genau den Wert hat, als würde er mit dem 4096-fachen der Farbträgerfrequenz getaktet.

Um ein Farbsignal zu identifizieren und den Quarzoszillator im TV zu synchronisieren, wird der ansonsten quadraturmodulierte Farbträger kurz nach dem Synchronimpuls für ca. 10 Perioden als sog. Burst gesendet, wobei die Phase von Zeile zu Zeile zwischen -135 und +135 Grad wechselt. Nach meinen Erfahrungen reichen aber auch 8 Perioden aus. Die Amplitude des Bursts beträgt mit +0,15V genau 50% der maximalen Amplitude bei maximaler Farbsättigung. Für die Grundfarben reicht eine feste Sättigung von 50% vollkommen aus, wodurch sich das Ganze noch weiter vereinfacht.

Das FBAS-Signal setzt sich aus dem Helligkeitssignal Y $[0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B]$ und den beiden Farbartsignalen U $[0,493 \cdot (B-Y)]$ und V $[0,877 \cdot (R-Y)]$ zusammen. Das Y-Signal wird direkt übertragen, mit den U- und V-Signalen wird der Farbträger quadraturmoduliert. Kurzgefasst wird das U-Signal mit dem Farbträger multipliziert und

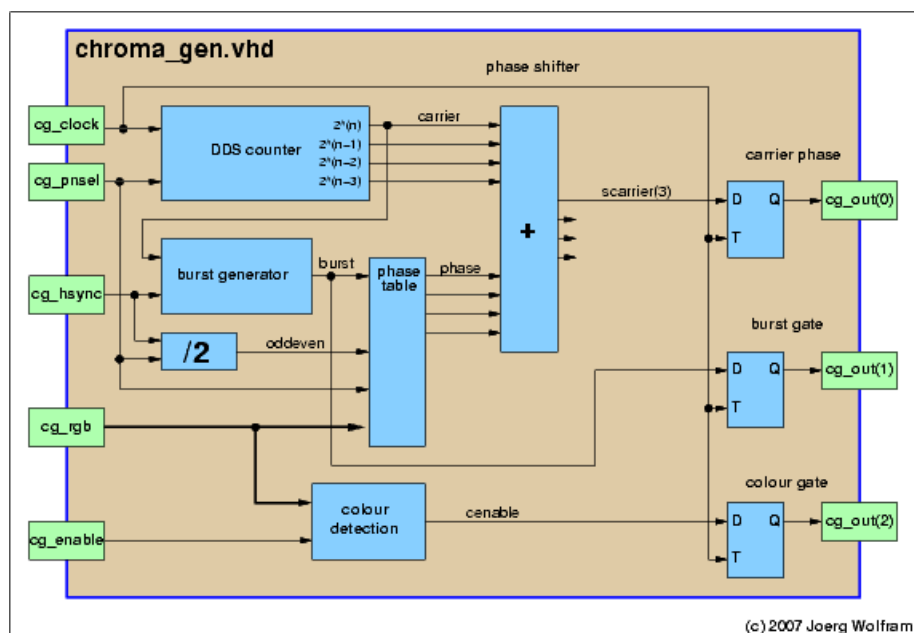
das V-Signal mit dem um 90 Grad oder 270 Grad phasenverschobenen Farbträger. Dabei wird der Farbton in der Phase und die Farbsättigung in der Amplitude des modulierten Farbträgers kodiert. Der modulierte Farbträger wird dann zum Y-Signal hinzuaddiert und fertig ist das FBAS-Signal. Naja, nicht ganz, während der Synchronimpulse muss es auf 0V gesetzt werden.

4 Die Realisierung

Nun stellt sich die Frage, wie man das alles in ein CPLD hineinbekommt. Der Farbträger und das Burst-Signal machen die wenigsten Probleme, die Modulation und Mischung mit dem Y-Signal ist schon weniger trivial. Nach einem Umweg über eine große Tabelle (in Version 0.21), die immer leichte Muster produzierte und auch sehr empfindlich gegen Hardwaretoleranzen war, habe ich den Encoder in Richtung Nachbildung des Signalpfades weiterentwickelt. Bei den ersten Versuchen zeigte sich, dass es theoretisch möglich ist, ein FBAS-Signal zu erzeugen, indem man den Signalpfad digital nachbildet. Allerdings ist der Logikaufwand recht hoch und das Problem mit den störenden Mustern war dadurch nicht wegzubekommen. In der vorliegenden Version sind nun die Signalpfade für Luminanz- und Chrominanz-Signal getrennt und die Addition erfolgt auf analogem Wege ausserhalb des CPLD's. Damit ist es auch möglich, das Chrominanzsignal tiefpasszufiltern, ohne dass die Bildschärfe leidet.

4.1 Chrominanzsignal

Die DDS-Synthese und Burst-generation entsprechen der Version 0.21, der Rest ist etwas anders realisiert. Die Komponente befindet sich in der Datei **chroma_gen.vhd**.



Aus dem Burst und den RGB-Signalen wird die Phasenverschiebung berechnet. Die Auflösung beträgt 4 Bit, was einer Schrittweite von 22,5 Grad entspricht.

- Der Burst hat eine Phasenlage von +135 und -135 Grad, das ergibt die Werte 6 bzw 10 (16-6)
- Blau hat bei geraden und ungeraden Zeilen eine Phasenlage von 0 Grad, da V=0 ist
- Die Werte für die anderen Farben kann man aus dem Farbkreis bestimmen, wobei die Phase zwischen den Zeilen zwischen n und 16-n wechselt.
- Bei NTSC ist die Burstphase und die Farbphase in allen Zeilen gleich

Die Phasenverschiebung wird nun zu den obersten 4 Bits des Farbträgers addiert, damit erhält man den modulierten Farbträger. Zur Vereinfachung wird nur ein Rechtecksignal erzeugt, damit wird nur das höchstwertige Bit des Ergebnisses der Addition benötigt. Für Schwarz, Weiss und Graustufen muß das Chrominanzsignal abgeschaltet werden, für das Burstsignal wird die Amplitude reduziert. Erweiterungen sind relativ einfach möglich, man kann mehr Farbtöne erhalten, wenn

mehr Bits zur Berechnung des Farbträgers herangezogen werden. Außerdem lassen sich relativ einfach verschiedene Sättigungsstufen realisieren.

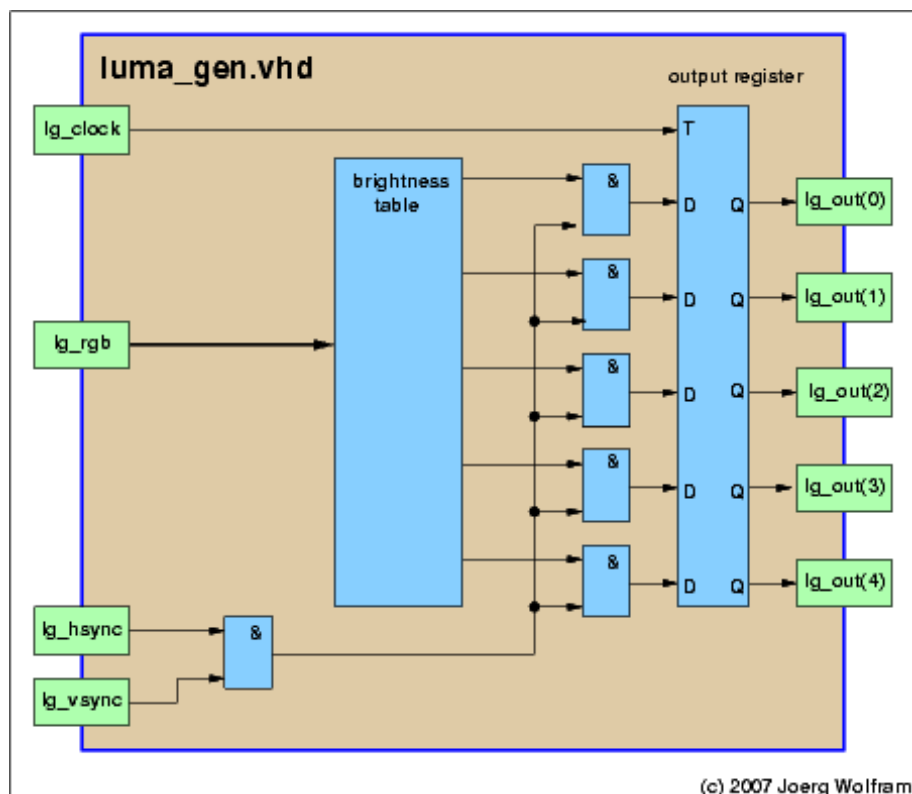
Da ich Probleme mit XST und der Konfiguration habe, wird die zu wählende Architektur nach der Component-Definition in der Datei **main.vhd** festgelegt. Für die verschiedenen Taktfrequenzen gibt es 4 Architekturen:

architecture	System	Takt	Bildqualität	CPLD-Typ
clock16	PAL	16 MHz	leichtes Kantenflimmern bei Farbübergängen	15ns
clock16	NTSC	16 MHz	teilweise unbrauchbar (Farben laufen aus)	15ns
clock20	PAL	20 MHz	leichtes Kantenflimmern bei Farbübergängen	10ns
clock20	NTSC	20 MHz	teilweise unbrauchbar (Farben laufen aus)	10ns
clock32	PAL	32 MHz	ruhiges Bild, leichtes Flimmern an Farbkanten	7ns
clock32	NTSC	32 MHz	rel. ruhiges Bild, leichtes Flimmern an Farbkanten	7ns
clock40	PAL	40 MHz	ruhiges Bild, leichtes Flimmern an Farbkanten	5ns
clock40	NTSC	40 MHz	ruhiges Bild, leichtes Flimmern an Farbkanten	5ns

Die Bildqualität habe ich auf einem LCD-TV getestet, PAL hat (systembedingt) die bessere Bildqualität. Timing ist aufgrund der nachgeschalteten Tiefpässe nicht so kritisch wie berechnet. (s.u.) Die 16- und 20 MHz Versionen eignen sich nur bedingt für NTSC, da bei Übergängen von bunt nach unbunt sowie unbunt nach bunt Verzögerungen auftreten, die als äusgewaschene Farbendeutlich sichtbar sind.

4.2 Luminanzsignal

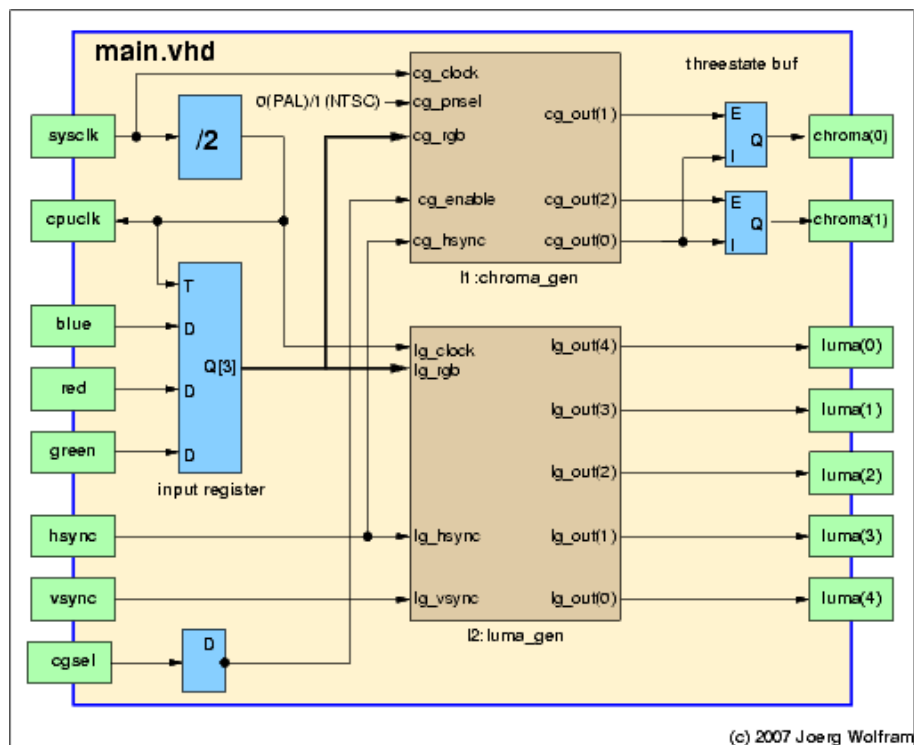
Das Luminanzsignal entspricht dem BAS-Signal bei Schwarzweiß-Darstellung. Die Komponente befindet sich in der Datei **luma_gen.vhd**.



Das Luminanzsignal wird über eine einfache Lookup-Tabelle erzeugt.

4.3 Die Beispielimplementierung

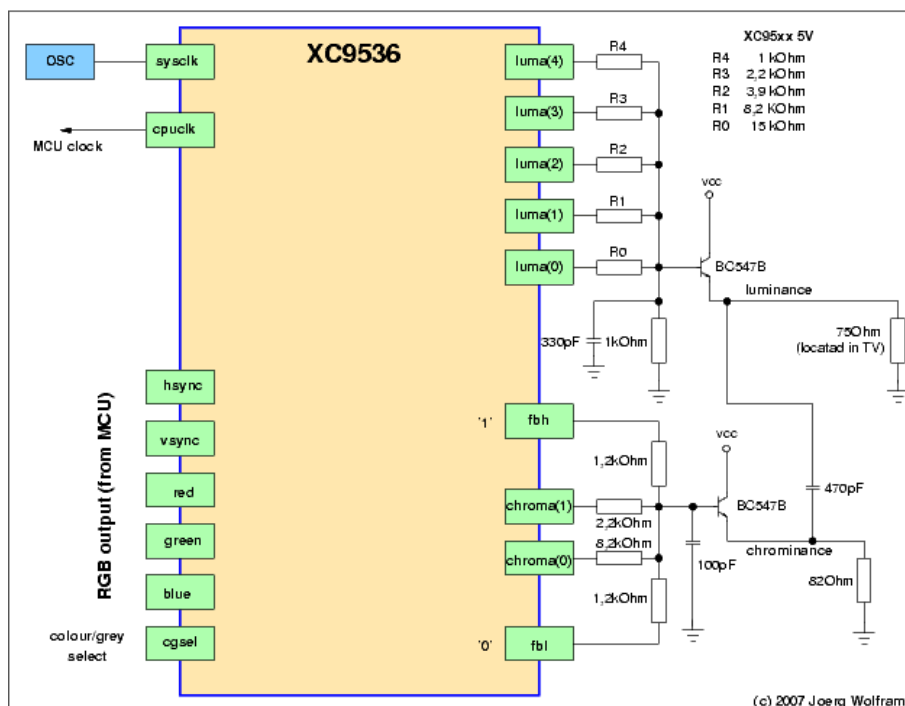
Um einen funktionsfähigen Chip zu erhalten, ist noch ein bisschen zusätzliche Logik notwendig:



Die RGB-Signale werden in einem Register zwischengespeichert, damit der Rest der Schaltung sauber arbeiten kann. Bei AVR-Mikrocontrollern hat sich z.B. gezeigt, dass die Zeit zwischen out PORTn und Signalwechsel am Port temperaturabhängig ist. Für das Chrominanzsignal sind zwei 3-State Ausgangsstufen vorgesehen und last but not least, der Vorteiler für die CPU-Frequenz. Die Luminanz-erzeugung wird nur mit der CPU-Frequenz getaktet, da Taktung mit dem Systemtakt in den Versuchen jede Menge Störungen erzeugt hat, die bis zum Abschalten der Farbdarstellung reichten. Außerdem ist eine schnellere Taktung als der CPU-Takt auch nicht nötig, da die CPU/MCU die Daten gar nicht schneller ausgeben kann.

4.4 Beschaltung des CPLD's

Die Beschaltung des CPLD's ist gegenüber der Version 0.21 ein ganzes Stück aufwändiger geworden.



Die gravierendste Änderung sind die zwei Transistoren am Ausgang. Diese werden in Kollektorschaltung betrieben und fungieren als Impedanzwandler. Im oberen Zweig wird das BAS-Signal erzeugt. Theoretisch ist es auch möglich, das BAS-Signal auch vom Mikrocontroller erzeugen zu lassen. Dabei ist aber enorm wichtig, dass das HSYNC-Signal kein Composite-Sync ist, damit die geraden/ungeraden Zeilen in der vertikalen Austastlücke korrekt weitergezählt werden. Ansonsten kann es passieren, dass der Trägeroszillator im TV jedesmal die Synchronisation verliert und das Bild erst (je nach TV) im unteren Drittel farbig ist. Im unteren Zweig wird das Chrominanz-Signal erzeugt. Die zwei 1,2kOhm Widerstände an den PINs **fbh** und **fbl** dienen zur Vorspannung in der Mitte des Signalhubs an den Ausgängen des CPLD's und vereinfachen so die Symmetrieeinstellung. Das Signal wird über einen Kondensator von 470-1000nF mit dem Luminanz-Signal gemischt, wie es auch bei einfachen Adaptern von S-Video auf FBAS gemacht wird. Das legt nahe, dass auf diese Weise auch ein S-Video Ausgang realisiert werden kann, getestet wurde es aber nicht.

5 Timing

Das Timing verlangt bei 32MHz nach einem 7ns CPLD und bei 40MHz nach einem 5ns CPLD, für den Privatgebrauch reicht nach meinen Erfahrungen aber auch (zumindest versuchsweise) ein 15ns Teil. Das liegt daran, dass kurze Impulse durch die Ausgangsschaltung gefiltert werden. Hat man mehr Makrozellen frei (39), so lässt sich das Timing noch entschärfen. Allerdings wurde nur das Verhalten bezüglich der Periodendauer des Signals **sysclk** bewertet, der Timing-Report gibt genauere Auskünfte und zeigt recht kritische Signalpfade bei der Chroma-Generation, bei denen selbst ein 5ns CPLD nur bis etwa 30MHz einwandfrei funktionieren sollte. Um das Ganze weiter zu entschärfen, könnte zum Beispiel die Chroma-Signalkette mit **sysclk** getaktete Zwischenregister bekommen (Pipelining), die dadurch entstehende Verzögerung des Chrominanz-Signals gegenüber dem Luminanz-Signal sollte zu vernachlässigen sein oder kann durch zusätzliche Register auch im Luma-Zweig kompensiert werden. Als Alternative gibt es noch die 16- und 20 MHz Versionen, bei denen es je nach TV an vertikalen Farbwechseln zu Störungen kommt.

6 Fazit

Das in VHDL geschriebene Design passt in ein preiswertes XC9536 CPLD, solange entweder nur PAL oder nur NTSC genutzt wird. Die in der Version 0.21 vorhandenen Bildstörungen wurden (zumindest bei der 32- und 40 MHz Version) praktisch eliminiert und das Design ist einfacher erweiterbar. Nachteilig ist der etwas höhere Schaltungsaufwand gegenüber der Vorgängerversion. Fertig compilierte Designs (nebst PIN-Beschreibung) sind im **jedec**-Ordner zu finden.

7 Changelog

18.2.2007 erste öffentliche Version (0.21)

- 16 und 20 MHz Version

9.3.2007 Version (0.31)

- Aufgabe der Tabellenvariante zugunsten einer flexibleren Lösung
- Übersichtlicheres Design, bessere Bildqualität und Erweiterbarkeit