VHDL Praktikum

Aufgabe 2: VGA-Signalerzeugung

Bertram, Alexander (B_TInf 2616, 6. Fachsemester)

Inhaltsverzeichnis

1. Aufgabenstellung

Es ist ein Design zu entwickeln, das ein VGA-Signal erzeugt und abhängig von zwei Werten je einen Balken auf einem Bildschirm auf und ab bewegt.

2. Bedienungsanleitung

2.1. Komponenten und ihre Funktionen

Auf der folgenden Grafik und in der darauf folgenden Tabelle sind die Komponenten und ihre Funktionen dargestellt. Die Zahlen hinter den einzelnen Funktionen bedeuten, dass diese Komponente für die Anzeige mit der entsprechenden Zahl zuständig ist.

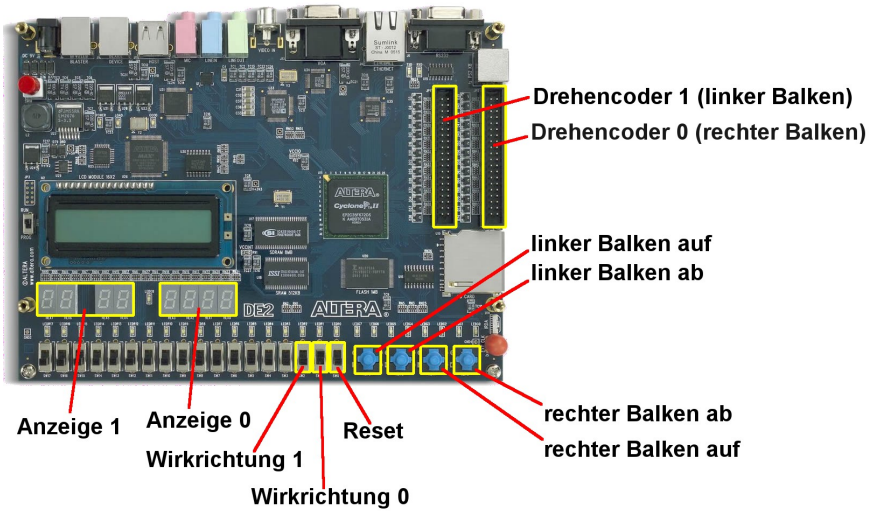

Abbildung 1: Übersicht der Komponenten

Komponente	Funktion
SW0	Reset
HEX0-3	Anzeige 0, vertikale Position des rechten Balkens
HEX4-7	Anzeige 1, vertikale Position des linken Balkens
KEY0	Rechten Balken ab bewegen
KEY1	Rechten Balken auf bewegen
KEY3	Linken Balken ab bewegen
KEY4	Linken Balken auf bewegen
JP2	Erweiterungsstecker für Drehencoder 0, Steuerung des rechten Balkens
JP1	Erweiterungsstecker für Drehencoder 1, Steuerung des linken Balkens
SW1	Wirkrichtungsschalter für den rechten Balken
SW2	Wirkrichtungsschalter für den linken Balken

Tabelle 1: Komponenten und ihre Funktionen

2.2. Reset

Sobald das Design auf das Board heruntergeladen wurde, sollte es zunächst resettet werden. Dies geschieht indem der Schalter SW0 in die obere und dann wieder in die untere Position bewegt wird.

2.3. Steuerung der Balken

Es gibt zwei Wege die Balken zu steuern.

Der erste Weg sind die Tasten KEY0 bis KEY3. Die Tastenbelegung ist der oberen Grafik bzw. Tabelle zu entnehmen.

Der zweite Weg sind die Drehencoder. Je nach Lage der Wirkrichtungsschalter werden die Balken auf und ab bewegt. Die verschiedenen Möglichkeiten sind der folgenden Tabelle zu entnehmen:

Schalterposition	Drehrichtung	Funktion
unten	im Uhrzeigersinn	Balken auf bewegen
unten	gegen den Uhrzeigersinn	Balken ab bewegen
oben	im Uhrzeigersinn	Balken ab bewegen
oben	gegen den Uhrzeigersinn	Balken auf bewegen

Tabelle 2: Kombinationen der Positionen der Wirkrichtungsschalter und der Drehrichtungen und ihre Funktionen

3. Entwicklungskonfiguration

3.1. Hardware

- Windows-kompatibler PC
- Altera DE2 Development & Education Board
- Zwei Drehencoder der Fachhochschule Wedel
- Monitor mit VGA-Anschluss

3.2. Software

- Quartus II 8.0 Web Edition
- ModelSim-Altera 6.1g

4. Designstruktur

Das Design wurde nach dem Top-Down-Prinzip entworfen und wird in diesem Abschnitt auch so beschrieben.

4.1. VGA-Signal: EVGASignal

4.1.1. Beschreibung

EVGASignal ist die Top-Level-Entity des gesamten Designs. Sie besteht aus drei Komponenten:

EClockDivider, EIOModul und EJoistsDraw. Die Komponenten werden verdrahtet und erzeugen die Signale, die für die Darstellung der Balken auf dem Monitor benötigt werden.

4.1.2. Designstruktur

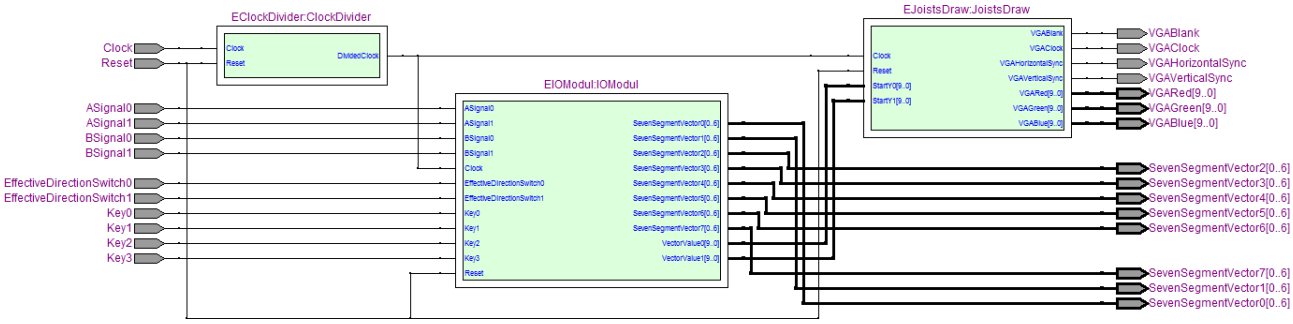

Abbildung 2: Designstruktur der Entity EVGASignal

4.1.3. Testbench: TBVGASignal

Diese Testbench bindet den Encodersimulator aus der ersten Aufgabe ein. Es werden zwei Klicks simuliert, so dass sich die Balken nach unten bewegen. Sie beinhaltet keine vernünftigen Testfälle, da die einzelnen Komponenten beim Entwickeln genug getestet wurden.

TBVGASignal eignet sich sehr gut, um die Signalverläufe sowohl in der funktionalen als auch in der post-synthese Simulation zu beobachten.

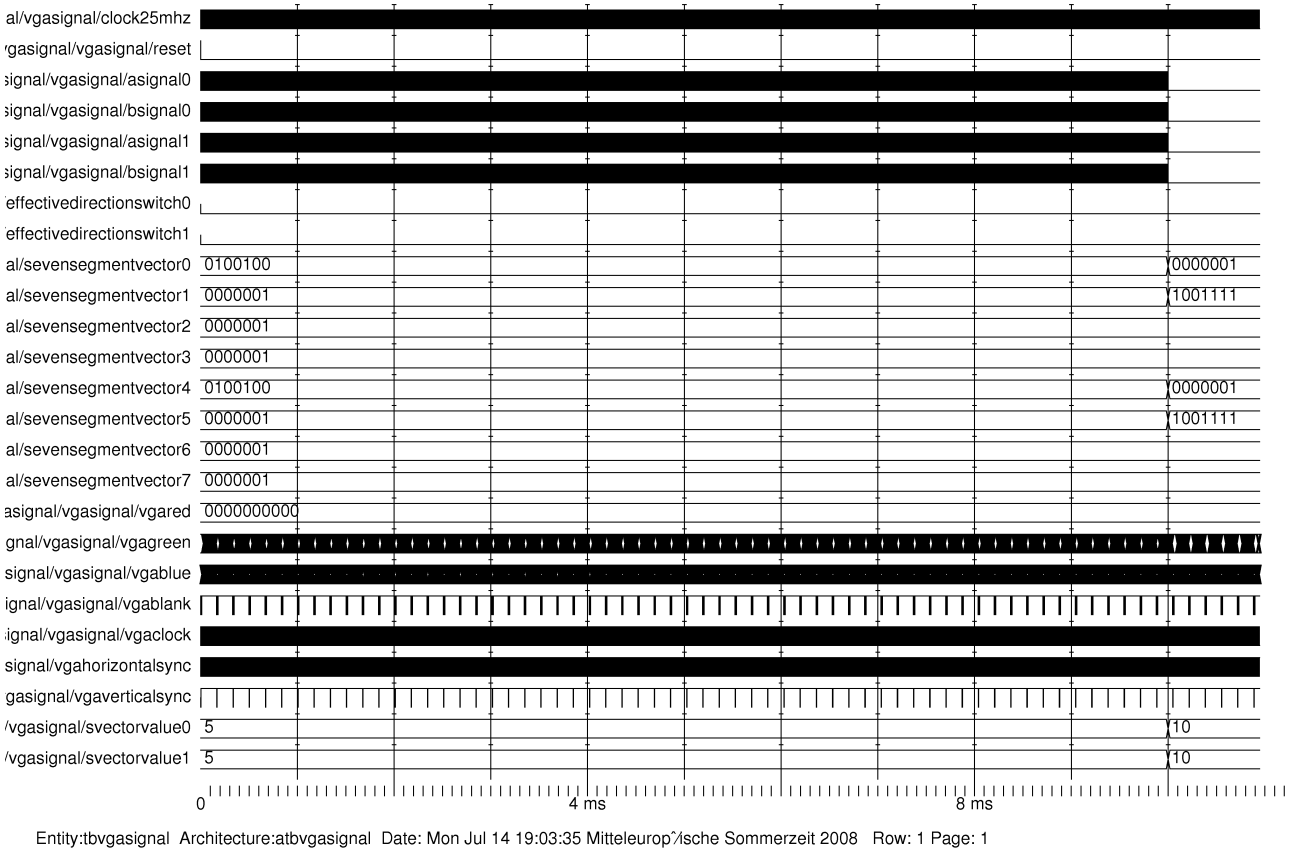

Abbildung 3: Auschnitt aus der Testbench TBVGASignal

4.2. Allgemeine Datentypen: Types

In der Package Types werden Datentypen und Konstanten definiert, die in mehreren Entities gebraucht werden. Dies erleichtert die Arbeit und schließt Fehlerquellen aus.

4.3. Frequenzteiler: EClockDivider

Die Komponente EClokDivider ist ein einfacher Frequenzteiler, der mit Hilfe eines Zählers realisiert wurde. Das Teilverhältnis wird bei der Instantiierung mit Hilfe eines Generics gesetzt.

4.4. Ein- / Ausgabemodul: EIOModul

4.4.1. Beschreibung

Diese Entity ist das Ergebnis der ersten Aufgabenstellung. Sie wurde bis auf einige kleinere Anpassungen unverändert übernommen.

4.4.2. Veränderungen

Zu den Veränderungen zählt z. B. die Umbenennung der Zählkomponente und der dazugehörigen Typen, Signalen und Variablen. Die Zeichenkette „Delay“ in den Namen wurde durch „Clock“ ersetzt. So heißt die Zählkomponente jetzt anstatt EDelayCounter EClockCounter. Ausserdem wurde das Zählverhalten des Zählers wegen eines gefundenen Bugs leicht modifiziert. Dieser zählt

jetzt einen Takt weiter, bis das Signal gesetzt wird, welches nach außen bekannt gibt, dass die Zählgrenze erreicht wurde.

Der gefundene Bug spielte in der ersten Aufgabe keine wesentliche Rolle, weil eine Taktperiode von 20 ns bezogen auf z. B. die Entprellungszeit von 5 ms vernachlässigbar klein ist.

Eine weitere Veränderung stellt die Anpassung der Zählgrenze der Entity EOutput dar. Da bei der Entwicklung der EOutput Generics verwendet wurden, mussten auch nur diese bei der Instanziierung verändert werden. Die Zählgrenze hängt von der Höhe des Bildschirms und der Balken ab, die durch Konstanten in der Types festgelegt werden.

4.4.3. Dokumentation

Für nähere Erläuterung wird hier auf die Dokumentation der ersten Aufgabe verwiesen.

4.5. Zeichnen der Balken: EJoistsDraw

4.5.1. Beschreibung

Diese Entity hat als Eingangssignale u. a. die von der Entity EIOModul erzeugten Werte. Am Ausgang werden Signale zur Ansteuerung eines VGA-Monitor erzeugt.

EJoistsDraw besteht aus mehreren Komponenten: EVGASync, EDrawPixel und EClockDivider. Diese werden instantiiert, sinnvoll verdrahtet und sind für die eigentliche Signalerzeugung zuständig.

4.5.2. Designstruktur

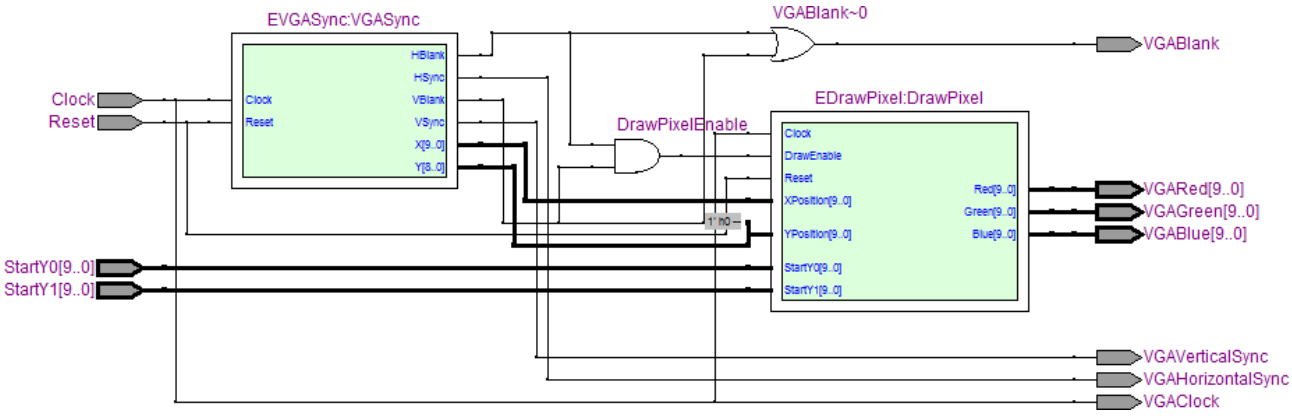

Abbildung 4: Designstruktur der Entity EJoistsDraw

4.6. Einführung in die VGA-Signalerzeugung

Ein Bild auf einem Monitor besteht aus vielen Zeilen, die wiederum aus Pixel bestehen. Das Bild wird Pixel für Pixel von oben links nach unten rechts aufgebaut. Für die Darstellung eines Pixels werden drei Farbwerte, rot, grün und blau, benötigt, die die Intensität der jeweiligen Farbe wiedergeben. Damit ein Pixel an der richtigen Stelle dargestellt wird, muss dem Monitor „mitgeteilt“ werden, wann eine Zeile bzw. ein Bild beginnt bzw. endet. Dies geschieht mit Hilfe von zwei Synchronisationssignalen: HSync und VSync. Durch das Anlegen bestimmter Pegel mit einem

bestimmten Zeitverhalten an diese Signale, kann der Monitor bestimmen, wann eine Zeile bzw. ein Bild beginnt bzw. endet.

4.6.1. Horizontale Synchronisation: Erzeugen einer Zeile

Für die horizontale Synchronisation ist das Signal HSync zuständig. Das Signalmuster sieht wie folgt aus:

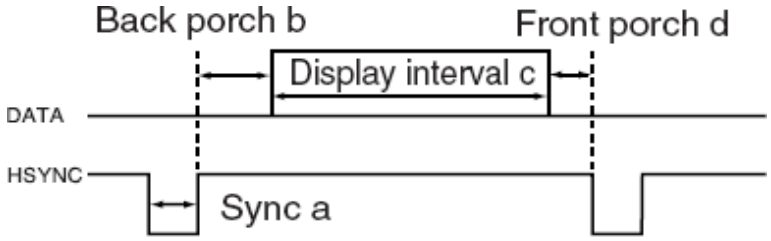

Abbildung 5: Signalmuster der Synchronisation

Durch einen Nullpegel, der die Zeit *a* dauert, signalisiert das Signal HSync das Ende einer alten und den Beginn einer neuen Zeile. Die Zeit *d* vor und die Zeit *b* nach dem Nullpegel müssen die Farbwertsignale und das Signal Blank auf Masse gelegt werden. Schließlich wird in der Zeit *c* die Zeile Pixel für Pixel mit einer bestimmten Frequenz ausgegeben.

VGA Modus		**Zeitverhalten in µs**				
Bezeichnung	**Auflösung (BxH)**	**a**	**b**	**c**	**d**	**Pixelfrequenz in MHz**
VGA (60 Hz)	640 x 480	3,8	1,9	25,4	0,6	25
VGA (85 Hz)	640 x 480	1,6	2,2	17,8	1,6	36
SVGA (60 Hz)	800 x 600	3,2	2,2	20	1	40
...	...	...	...	...	...	...

Tabelle 3: Zeitverhalten der horizontalen Synchronisation

Der Tabelle ist zu entnehmen, dass die Bildauflösung und die -wiederholfrequenz lediglich von der Länge der Impulse und der Pixelfrequenz abhängt.

4.6.2. Vertikale Synchronisation: Erzeugen eines Bildes

Analog zur horizontalen Synchronisation läuft die vertikale Synchronisation ab, für die das Signal VSync zuständig ist.

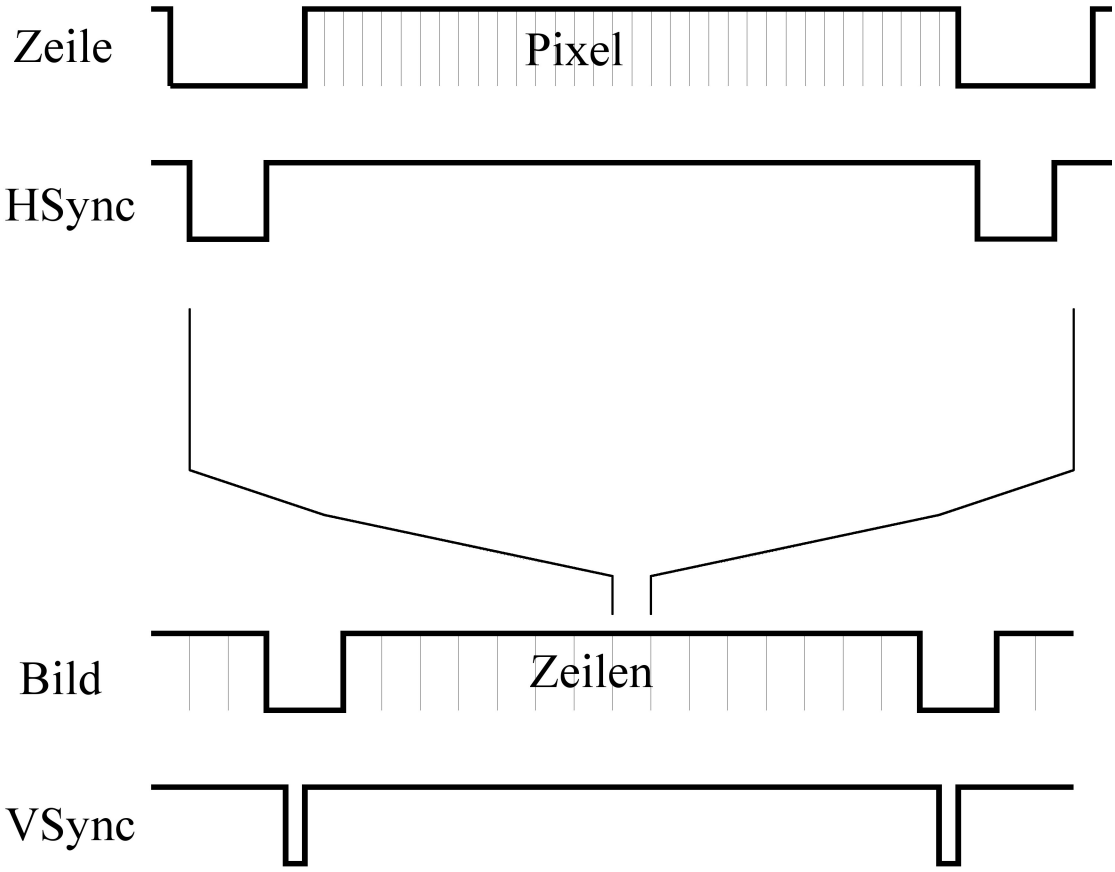

Abbildung 6: Zusammenhang zwischen einer Zeile und einem Bild

VGA Modus		Zeitverhalten in Zeilen			
Bezeichnung	**Auflösung (BxH)**	**a**	**b**	**c**	**d**
VGA (60 Hz)	640 x 480	2	33	480	10
VGA (85 Hz)	640 x 480	3	25	480	1
SVGA (60 Hz)	800 x 600	4	23	600	1
...	...	...	...	...	...

Tabelle 4: Zeitverhalten der vertikalen Synchronisation

4.7. VGA-Synchronisation: EVGASync

4.7.1. Beschreibung

Die Entity EVGASync erzeugt eine Impulsfolge, wie sie weiter oben in der Einführung beschrieben wird. Zusätzlich berechnet sie die momentane Position auf dem Monitor.

4.7.2. Aufbau und Funktionsweise

Die Architektur der Entity besteht aus zwei Zählern, einem Pixel- und einem Zeilenzähler.

Erreichen die Zähler bestimmte Werte, die per Generics bestimmt werden können, werden entsprechende Ausgangssignale zur Synchronisation des Monitors gesetzt.

Zusätzlich wird in dieser Entity die aktuellen Koordinaten bestimmt, die für die Darstellung nötig sind.

4.7.3. Testbench: TBVGASync

In der Testbench wird mit Hilfe von Asserts das Zeitverhalten für die VGA-Synchronisation sowohl in die horizontale als auch in die vertikale Richtung funktional getestet.

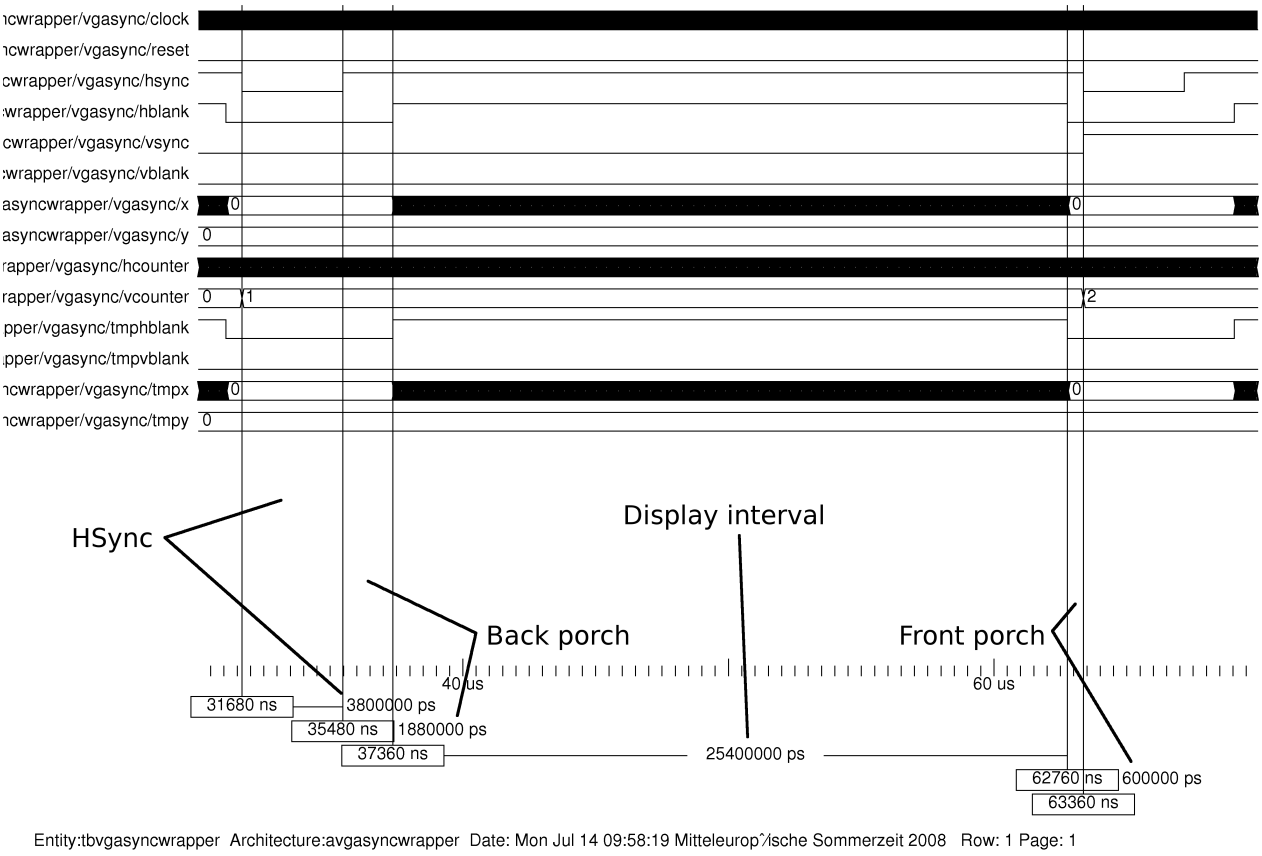

Abbildung 7: Horizontale Synchronisation

Wie in der Abbildung zu sehen ist, ist das „Back porch“ Interval 18,8µs lang statt 19µs. Das liegt an der Betriebsfrequenz von 25MHz, mit der ein Intervall von 19µs nicht erzeugt werden kann. Dies kann vernachlässigt werden, da durch diesen „Fehler“ im erzeugten Bild lediglich einige Pixelspalten fehlen.

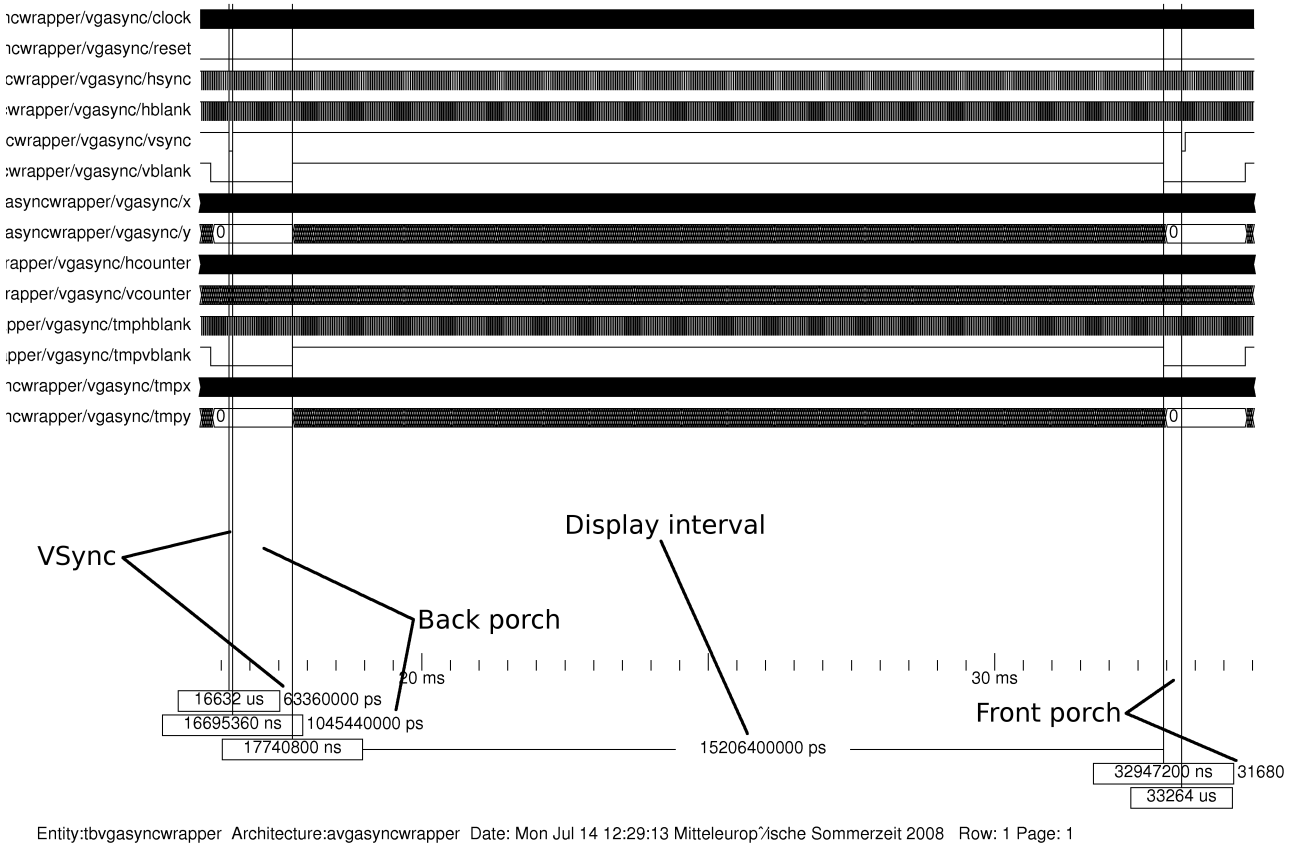

Abbildung 8: Vertikale Synchronisation

4.8. Pixel zeichnen: EDrawPixel

4.8.1. Beschreibung

Diese Entity berechnet die Farbe des Pixels an der aktuellen Position und setzt dementsprechend die Farbwertsignale. Für die Berechnung sind, neben den aktuellen Koordinaten auf dem Monitor, die Höhe und Breite der beiden Balken erforderlich. Die Koordinaten liegen als Eingangssignale an und die Farben für die beiden Balken und den Hintergrund, sowie die Maße der Balken, werden bei der Instantiierung über Generics gesetzt.

4.8.2. Testbench: TBDrawPixel

Diese Testbench testet, ob die Farben, die die Entitiy EDrawPixel berechnet, richtig sind.

Es werden zunächst die Farben an den Übergängen zwischen den Kanten der Balken und dem Hintergrund getestet. Dann werden die Balken bewegt und es wird wieder der gleiche Test durchgeführt.

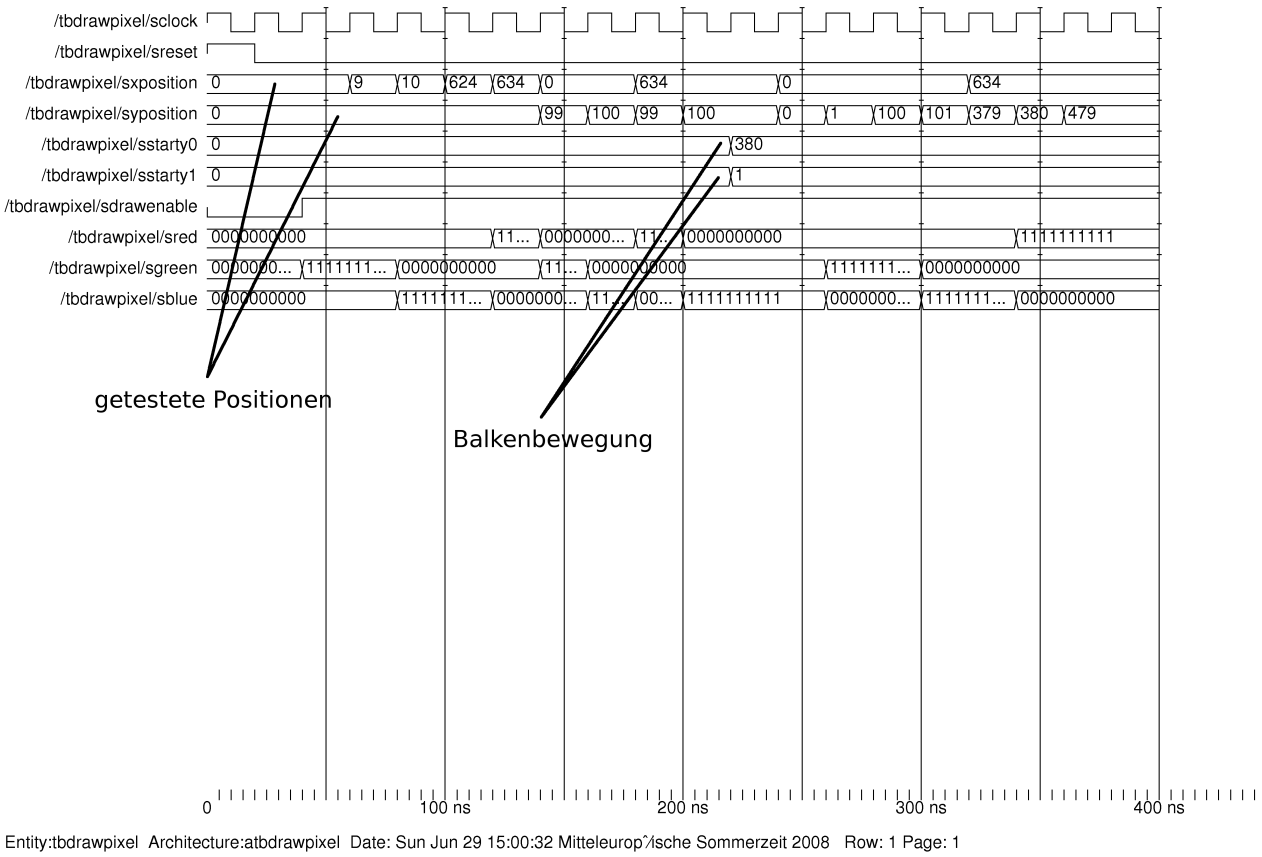

Abbildung 9: Signale, die von der Entity EDrawPixel erzeugt werden