Laborübung 3

Die Bearbeitung der Aufgaben erfolgt in Gruppen zu je max zwei Teilnehmern. Alle Aufgaben sind schriftlich zu lösen und in einem Protokoll festzuhalten. Alle praktische Realisierungen sind dem Dozenten im Rahmen der Gruppenrücksprache zu präsentieren. Erfolgreiche Abnahmen werden auf dem Aufgabenblatt schriftlich quittiert. Eine erfolgreiche Laborteilnahme ist Voraussetzung für die Teilnahme an der Klausur am Ende des Semesters. Labore werden benotet und gehen zu 30% in die Endnote ein. Bitte bedenken Sie, dass eine gute Vorbereitung zu den Laborterminen unabdingbar ist.

Vorbereitung

Beschäftigen Sie sich mit dem VGA Standard. Wofür sind die Front und Back Porch?

Lesen Sie sich die VGA Vorlage durch und vollziehen Sie das Konzept des Zwischenspeichers im FPGA nach. Wie viel Speicher benötigt der Zwischenspeicher auf dem FPGA?

Beschreiben Sie die in der Vorlage vorgegebenen Prozesse.

Übersicht VGA

VGA ist ein Bildschirmstandard, welcher noch aus der Zeit stammt, in dem ein Elektronenstrahl die Pixel auf einem Bildschirm beleuchtet hat. Dieser Elektronenstrahl bewegte sich von oben links nach rechts, um alle Pixel der Zeile auszuleuchten. Anschließend wurde er wieder nach links gebracht, um die nächste Zeile auf den Bildschirm zu zeichnen. Nach der letzten Zeile wurde er wieder nach oben geführt und das nächste Bild wurde gezeichnet.

Die Signale von VGA bestehen aus der VGA-Clock, den 3 Farbsignalen für Rot, Grün und Blau und den 2 Signalen für die vertikale und horizontale Synchronisation.

Eine neue Zeile beginnt, wenn das VSYNC Signal auf Low geht. Wenn das VSYNC Signal wieder auf High geht, beginnt die Back Porch, anschließend folgt das Aktive Videosignal und die Front Porch.

Das Synchronisationssignal ist also nur während des Synchronisationspulses auf Low, die restliche Zeit ist es High. Die Farbausgänge sind nur während des aktiven Videos aktiv, während der Synchronisation, der Front und der Back Porch werden sie auf Null gezogen.

Horizontale Synchronisation sieht genauso aus wie die vertikale: auch hier gibt es ein Synchronisationssignal, die Back Porch, das aktive Signal und die Front Porch.

Um die Implementierung der VGA-Ansteuerung zu erleichtern, wird folgende logische Aufteilung vorgenommen:

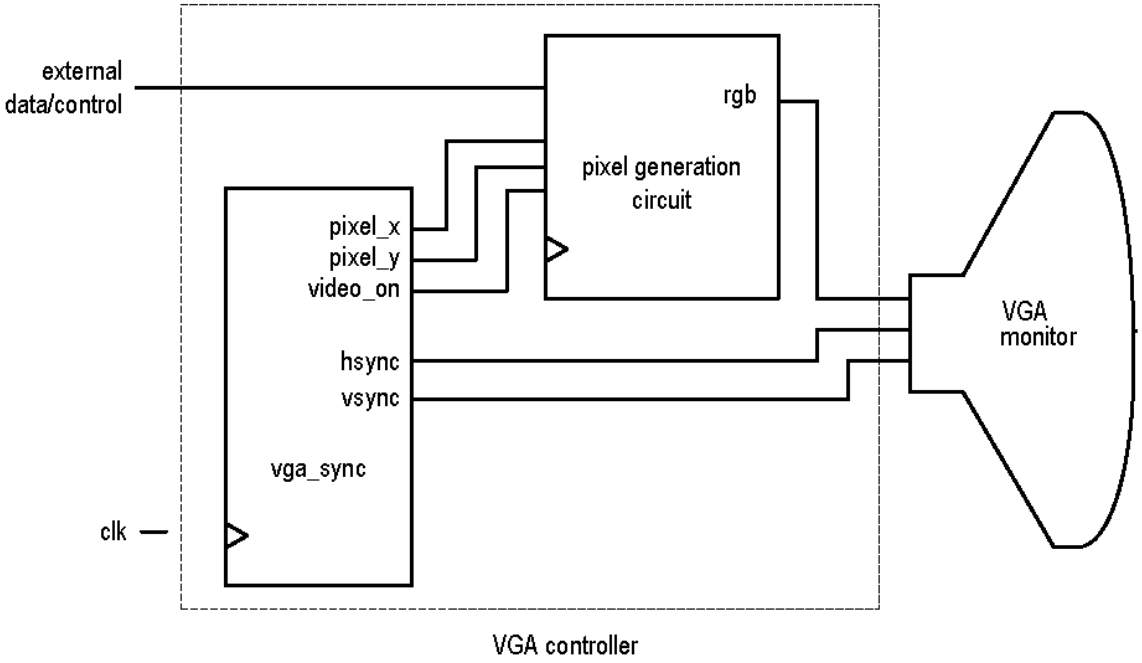

Abbildung 1: Aufteilung der logischen Komponenten

In der Vorlage finden Sie die AXI-Ansteuerung, welche den externen Datenkontrollteil übernimmt, sowie ein paar Prozesse, welche aus dem Zwischenspeicher die Pixel auslesen und damit die Aufgabe der *pixel_generation_unit* übernehmen.

Ihre Aufgabe wird es nun sein, die Komponente zu implementieren, welche in diesem Bild als *vga_sync* bezeichnet ist.

Aufgabe 1

Legen Sie ein neues Projekt in Vivado an. Im *Project Manager* klicken Sie auf *Add Sources*. Anschließend klicken Sie oben links auf das Plus und dann auf *Create File*. Wählen Sie VHDL als Sprache aus und geben Sie der Datei den Namen *vga_sync.vhd*. Achten Sie darauf, dass der Dateiname und der Name der Entity gleich sein müssen.

Anschließend können Sie im nächsten Fenster im GUI, oder wenn Sie dieses schließen direkt im Code ihre Ein- und Ausgänge definieren.

Um den VGA Block zu bauen, beginnen Sie als erstes mit 2 Zählern, einem für die horizontale und einem für die vertikale Synchronisation. Verwenden Sie für diese Zähler bitte den Datentyp Unsigned.

Zähler horizontal:

- zählt von 0 bis 799

- zählt bei jeder steigenden Taktflanke der VGA Clock
- Sync Pulse von 656 bis 751
- aktives Video von 0 bis 639 (Blanking ab 640)

Zähler vertikal:

- zählt von 0 bis 524
- zählt hoch, sobald horizontaler Zähler = 656 ist
- Sync Pulse von 490 bis 491
- aktives Video von 0 bis 479 (Blanking ab 480)

Nützliche Konstanten für die Implementierung:

```
CONSTANT h_active_video: INTEGER := 640; -- horizontal active video
CONSTANT h_front_porch: INTEGER := 16; -- horizontal front porch
CONSTANT h_retrace: INTEGER := 96; -- horizontal retrace / sync pulse length
CONSTANT h_back_porch: INTEGER := 48; -- horizontal back porch

CONSTANT v_active_video: INTEGER := 480; -- vertical active video
CONSTANT v_front_porch: INTEGER := 10; -- vertical front porch
CONSTANT v_retrace: INTEGER := 2; -- vertical retrace / sync pulse length
CONSTANT v_back_porch: INTEGER := 33; -- vertical back porch
```

Um zu testen, ob die Zähler funktionieren, bauen Sie eine Testbench, setzen Sie diese bei den Quellen mit einem Rechtsklick als Top Level Entity und starten Sie die Simulation mit im *Flow Navigator* unter *Simulation* → *Run Simulation* → *Run Behavioral Simulation*.

Als nächstes implementieren Sie die HSYNC und VSYNC Signale. Des Weiteren implementieren Sie die das VIDEO ON - Signal, was immer dann 1 ist, wenn gerade aktives Video ist, beziehungsweise der horizontale oder vertikale Zähler nicht im Blanking Bereich sind.

Für die Implementierung dieser Signale empfiehlt es sich, die Reihenfolge etwas anders zu betrachten, also mit dem aktiven Video zu beginnen und danach Front Porch, Sync und Backporch kommen zu lassen. Dies ermöglicht nämlich die direkte Ausgabe des Counters als aktuellen Pixels zur Adressberechnung des RAMs, ohne, dass man von diesen Counter noch ein Offset abziehen muss. Das ist auch der Grund, warum der vertikale Zähler bei 656 hochzählt, das ist dann nämlich genau nach aktivem Video und Front Porch, also genau zu Beginn des horizontalen Sync-Pulses / Retraces.

Überprüfen Sie Ihren Code mit einer Simulation einer Testbench.

Wenn ihr Code nun funktionsfähig ist, können Sie diesen in einen IP Core migrieren. Legen Sie dazu einen neuen AXI Slave IP Core an. In der Vorlage ist bereits eine Port Map für den *vga_sync* Block.

Nach erfolgreicher Synthese und Implementation können Sie Ihren VGA IP Core fertigstellen. (Überspringen Sie die Synthese und Implementation beim Erstellen des IP Cores nicht. Sollte einer der beiden Schritte fehlschlagen und Sie haben dies nicht beim Erstellen des IP Cores getestet, so wird dieser Schritt beim Block Design fehlschlagen.)

Erstellen Sie ein Blockdesign, in welchem Sie den von Ihnen erstellten VGA IP Core einbinden, sowie einen GPIO Block zum Einlesen der 8 Switches.

Ihr IP Core benötigt noch ein 25,175 MHz Clock Signal und ein Reset. Um das Clock Signal zu erzeugen, instanziieren Sie einen *Clocking Wizard IP* in Ihrem Block Design. Der *clk_in* Eingang wird mit mit dem *FCLK_CLK0* Ausgang von dem Zynq verbunden, anschließend können Sie mit einem Doppelklick auf den Clocking Wizard die gewünschte Frequenz einstellen. Den Reset erzeugt man am Besten, indem man den *Utility Vector Logic IP* als Inverter konfiguriert und damit das *locked* Signal von dem Clocking Wizard invertiert. Der Clocking Wizard bietet auch eine *Connection Automation* an, welche den *reset* Eingang als Pin anlegt. Diesen müssen Sie aber noch mit dem constraints file auf einen der 5 Push Buttons legen, damit das Design synthetisiert und implementiert werden kann.

Schreiben Sie nun im SDK eine Anwendung, um Ihr Design mit Ihrem VGA Block zu testen.

Sollte sich Ihr VGA Block als fehlerhaft erweisen, können Sie ihn nochmal editieren, wenn Sie einen Rechtsklick auf ihn im Blockdesign machen und dann auf *Edit in IP Packager* klicken.

Aufgabe 2

Jetzt können Sie kreativ werden: Schreiben Sie ein Programm, was Rechtecke und andere mathematische Formen auf dem Display anzeigt. Diese sollen über die Schalter steuerbar sein.

Aufgabe 3 (Zusatz)

Wem 4 verschiedene Farben bei der Ausgabe nicht ausreichen, kann den VGA IP Core auf 12 Bit Farben umschreiben.

Nur vom Dozenten nach erfolgreicher Abnahme auszufüllen!

Gruppenname

Abnahme

Aufgabe 1	Aufgabe 2	Aufgabe 3

Beurteilung

Kriterium	Max.	Punkte
Themenbearbeitung entsprechend Zielstellung	15	
Fundierte Erarbeitung des Lösungsansatzes	20	
Qualität der technischen Umsetzung	15	
Funktionsfähigkeit der technischen Umsetzung	20	
Darstellung von Ergebnissen und Lösungsvorschlägen	15	
Bearbeitungszeit	15	
Gesamtpunkte für Beurteilungskriterium	100	
Note		

Index	95	90	85	80	75	70	65	60	55	50
Note	1,0	1,3	1,7	2,0	2,3	2,7	3,0	3,3	3,7	4,0