

Pong auf einem FPGA (Nexys 4) mit Unterstützung für Steuerung über ein Kameramodul (OV7670).

name,name,name,name

June 15, 2022

Contents

1 Einleitung

Die folgende Ausarbeitung ist eine kleine Einführung in die Welt der FPGAs, von einer einfachen Farbausgabe auf einem Monitor bis hin zu einem Videospiel, das über eine Kamera gesteuert werden kann.

1.1 Ziele und Plan des Projekts

Das Ziel dieses Projekts war der FPGA-Entwicklung zu verstehen und zu erforschen und gleichzeitig praktische Erfahrungen zu sammeln. Die obige Definition ist eher abstrakt, daher wurde eine Reihe von Zielen festgelegt. Im Folgenden sind die Ziele aufgelistet:

- FPGAs und VHDL verstehen
- Anzeigen einer einzelnen Farbe auf einem VGA-Monitor
- Entwicklung eines einfachen Spiels (Pong)
- Das Spiel durch die Kamera steuern, um es weiter zu vertiefen

1.2 Field-Programmable Gate Arrays

FPGAs sind moderne programmierbare logische Geräte, die so konfiguriert werden können, dass sie jede logische Operation ausführen. Ein FPGA enthält typischerweise eine Matrix aus programmierbaren Elementen, auch bekannt als Configurable Logic Blocks (CLBs). CLBs enthalten LookUp Tables (LUTs), die als Logik- oder Speicherelemente verwendet werden können. Die Konfigurationsdaten werden im Speicher abgelegt.

1.3 VHDL

Die Abkürzung VHDL steht für Very High-Speed Integrated Circuit Hardware Description Language. VHDL ist eine Hardwarebeschreibungssprache, eine Sprache zur Beschreibung von digitalen elektronischen Systemen.

2 VGA

VGA (Video Graphics Array) ist ein analoger Videostandard, der hauptsächlich in Personal Computern verwendet ist.

2.1 Wie funktioniert ein VGA?

Beim VGA handelt es sich um einen digitalen Übertragungsstandard für Bildsignale. Das Bild wird Zeilenweise übertragen und dabei hat jedes Pixel eine fest definierte Dauer. Die Farbinformationen eines Pixels liegen als analoge Spannungen an und werden bei jedem Pixel neu eingelesen. Sobald eine Zeile zu Ende ist, wird ein Horizontal-Synchronisationssignal ausgelöst, wodurch der Monitor die Zeile wechselt und wieder an den Anfang einer Zeile springt. Wenn alle Zeilen durchlaufen sind, löst der VGA-Controller ein Horizontal-Synchronisationssignal aus, wodurch der Monitor wieder an die Anfangsposition springt. Der sichtbare Teil des Bildschirms hat eine Auflösung von 640 x 480 Pixel. Der Monitor hat auch Bereiche, die nicht sichtbar sind, aber für die Synchronisation benötigt sind. Mit diesen Teilen beträgt die Gesamtbreite des Bildschirms 800 Pixel, und die Gesamthöhe beträgt 524 Pixel. Im Folgenden findet sich eine kurze Beschreibung der einzelnen Teile des Bildschirms:

- **Active Video:** Das ist der sichtbare Teil des Bildschirms, die Videoausgabe ist aktiviert.
- **Front Porch:** Der Data Output zum Monitor muss ausgeschaltet sein für eine bestimmte Zeit, die als "Front Porch" bezeichnet wird, bevor der HSync-Pulse stattfinden kann.
- **Sync Pulse:** Signalisiert das Ende einer Zeile und den Beginn der nächsten Zeile
- **Back Porch:** Das ist der Teil, der vor dem Start des Active Videos beginnt

Clock	Active video	Front Porch	Sync Pulse	Back Porch
Horizontal	640	16	96	48
Vertical	480	11	2	31

2.2 Verbindung zwischen FPGA und VGA

Der VGA-Anschluss hat insgesamt 15 Pins, aber nur 5 Signale werden für den Gebrauch benötigt:

- **HSYNC** - horizontale Synchronisierung Signal. Dieses Signal gibt die horizontale Position des aktiven Pixels
- **VSNC** - vertikale Synchronisierung Signal. Diese Signal gibt die vertikale Position des aktiven Pixels
- **RED** - roter Farbkanal
- **GREEN** - grüner Farbkanal
- **BLUE** - blauer Farbkanal

Code Implementation:

Für den VGA-Controller werden zwei Hauptelemente benötigt: ein 25MHz-Clock und Counters (für die vertikalen und horizontalen Pixelkoordinaten). Der horizontale und der vertikale Counter zählen bis 799 bzw. 525. Diese Counters werden im Takt des Clocks aktualisiert. Wenn der horizontale Counter zwischen 48 und 687 liegt und der vertikale Counter zwischen 33 und 512 liegt, ist das active Signal true, und somit die x-y-Pixelsignale auf dem Bildschirm angezeigt werden können

3 Pong

Das Pong-Spiel besteht aus einem Ball, der auf dem Bildschirm hin und her springt und 2 Paddeln (die sich nur vertikal bewegen können) mit welchen die Spieler den Ball wieder aufspringen lassen können. Der Ball muss sich auf dem Bildschirm bewegen und zurückspringen, wenn der Ball ein Objekt (eine Grenze oder ein Paddle) berührt.

Game Rules:

- Es gibt zwei Paddles, die sich nach oben und unten bewegen
- Die Paddles können nicht über die horizontalen Kanten des Bildschirms hinausgehen
- Der Ball startet in der Mitte und bewegt sich in eine diagonale Richtung
- Der Ball springt von den Paddles und den horizontalen Seiten des Bildschirms ab
- Wenn der Ball eine vertikale Seite berührt. Es wird ein Punkt vergeben, der Ball wird zurückgesetzt

3.1 Paddle Logic:

Diese Pong-Spiel ist ein Zwei-Personen-Spiel, daher werden zwei Paddles benötigt. Abmessungen und Abbildungen sind für beide Paddles gleich, aber sie haben verschiedene Positionskoordinaten. Die Paddle Steuerung ist recht einfach - sie reagiert auf Benutzereingaben, indem sie die Koordinate jedes Paddles erhöht oder verringert. Wenn ein Paddle am oberen oder unteren Grenze steht, ist keine weitere Bewegung in diese Richtung erlaubt.

3.2 Ball Logic:

Der Ball startet in der Mitte und bewegt sich in eine diagonale Richtung. Der Ball springt auf, wenn es ein Paddle oder eine horizontale Seite berührt und wenn es eine vertikale Seite berührt ein Punkt der Spieler, der gegenüber der berührten Seite steht, bekommt einen Punkt. Der Ball hat 2 Richtungen, X und Y. X und Y können entweder negativ oder positiv sein. Der Ball hat nur 2 Objekte, die es berühren und von denen es abspringen kann: ein Paddle oder eine horizontale Kante. Wenn der Ball auf eine horizontale Kante trifft, bleibt die x-Richtung unverändert, aber die y-Richtung wird invertiert. Wenn der Ball an einem Paddle aufspringt, bleibt die y-Richtung gleich, aber die x-Richtung wird invertiert.

(+ bilder mit richtungen)

3.3 Score Logic:**3.4 Game start end?****4 Kamera****4.1 Wie funktioniert ein Kamera?****4.2 Verbindung zwischen Kamera und FPGA****4.3 Testbild?****4.4 Farberkennung****5 Alle zusammen(VGA+PONG+KAMERA) ?****5.1 VGA+PONG****5.2 PONG+Kamera****6 Fazit**