

Zuerst überlegten wir uns, wie wir das Spiel aufbauen wollten und welche Komponenten notwendig waren. Dabei gab es noch das Problem, dass unser VGA-Kontroller aus der vorherigen Aufgabe nicht funktioniert hat. Entsprechend mussten wir also gleichzeitig die Spielelogik und den VGA-Kontroller entwickeln.

Anhand der in den Aufgaben geforderten Funktionalitäten erstellten wir nun einen Entwurf. Wir wollten (wie schon in der vorherigen Aufgabe) nebst eines Signalgenerators auch einen Pixelgenerator. Dieser sollte aber unabhängig von der Spielelogik sein. Dementsprechend brauchten wir noch ein Logik-Modul.

Signalgenerator:

Als Signalgenerator konnten wir den bisher erstellten – bzw. neu erstellten – wiederverwerten.

Pixelgenerator:

Hier überlegten wir uns, welche Komponenten in unserem Pong-Spiel auftreten sollten. Nebst eines notwendigen Balls, Schläger und Wänden wollten wir auch dem Spieler eine Punkteanzeige anbieten. Ball und Schläger ließen sich leicht erzeugen, da wir beide mit einem Rechteck approximieren konnten. Die Wänden wurden zum Großteil schon in der vorherigen Aufgabe erzeugt (WhiteBorder), wir mussten nur eine der Wandbedingungen (links) entfernen.

Die Punkteanzeige erwies sich als schwieriger. In einem ersten Versuch wollten wir, wie bei den anderen Pixelgeneratoren, alles hintereinander schalten. Dadurch hätte immer die letzte Ziffer der Punkte abgearbeitet werden können. Allerdings wurden dazu dynamische Ganzzahldivisionen und Modulooperationen benötigt, die von Xilinx nicht unterstützt werden. Da wir kein eigenes Divisionsmodul schreiben wollten, arbeiteten wir unseren Entwurf um.

Wir führten einen neuen Typ ein, der jeweils 4 Zahlen zwischen 0 und 9 beinhalten konnte. Dadurch hatten wir ein Zahlenarray, auf das man direkt zugreifen konnte. Nun musste nur bei der Punkteberechnung etwas mehr Aufwand betrieben werden.

Spielelogik:

Die eigentliche Logik eines Pong-spieles ist sehr einfach und lässt sich ebenso simpel implementieren. Wie schon im Pixelgenerator beschrieben, mussten wir hier nur auf die Punktegenerierung aufpassen, das diese unserem neuen Typ genügt.

Ein weiteres Problem war, dass wir nicht über mögliche Geschwindigkeiten nachgedacht hatten. Dies führte dazu, dass in einem ersten Probelauf die Zeit direkt von der VGA-Clock des FPGAs genommen wurde. Dies war allerdings viel zu schnell. Anstelle der VGA-Clock nahmen wir nun das VSYNC-Signal; Damit wird in jedem Frame ein Aktualisierungsschritt durchgeführt.

Belegte Knöpfe:

DIP_01: Hardcore-Mode

Button_left: Paddle_UP

Button_right: Paddle_DOWN

Button_rst: Resets the game

Als alles zusammengefügt war, ergab sich noch ein Problem: Bei der Signalausgabe kommt ein Flackern in das Bild. Allerdings waren wir nicht in der Lage, den Fehler zu lokalisieren.