

# Vorlesung Architekturen und Entwurf von Rechnersystemen



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Wintersemester 2018/2019

Übungsblatt 6

Prof. Andreas Koch, Jaco Hofmann

Da nun in den Übungen 4 und 5 zusammen der passende Filter implementiert wurde, richtet diese Übung ihre Aufmerksamkeit auf die Speicherung aller derer Pixel, die nicht für den aktuellen Filtervorgang von Nöten sind.

---

## Aufgabe 6.1 Simpler Puffer

Da der FAST-Algorithmus viele Pixel gar nicht braucht, sollen in dieser Teilaufgabe 2 Arten von Puffern entwickelt werden: "Kleinere", die mit normalen FIFOs arbeiten, und "Größere", die mit BRAMFIFOs arbeiten, da zwischen je zwei Reihen von Pixeln (Bildbreite - 7) Pixel gespeichert werden müssen und das zu viele für herkömmliche FIFOs sind. Dank der Interface-Modul-Trennung von Bluespec lässt sich dies leicht umsetzen. Entwerfen Sie 2 Module, `mkBuffer` und `mkBRAMBuffer`, die eine angegebene Anzahl von Werten speichern, bevor sie (sinnvolle) Werte wieder ausgeben.

---

## Aufgabe 6.2 Testen der Implementierung

Testen Sie Ihre Implementierung mit einer herkömmlichen Testbench. Dabei ist es Ihnen überlassen, ob der Speicher, wenn er nicht voll ist, gar nichts zurück gibt oder ein `tagged Invalid`.

---

## Aufgabe 6.3 Entwurf von komplexeren Speichern

Im späteren Modul müssen jeweils 6 Reihen von Pixeln im BRAM gespeichert werden, nicht wieder gebrauchte Pixel können verworfen werden. Entwerfen Sie dafür ein Modul, welches 6 Werte gleichzeitig annimmt und auch wieder ausgibt. Auch dieses Modul soll über die Anzahl der Pixel, die vorgehalten werden sollen, parametrisierbar sein.

---

## Aufgabe 6.4 Testen der Implementierung

Testen Sie Ihre Implementierung der komplexeren Speicher.