

Übung zur Vorlesung Architekturen und Entwurf von Rechnersystemen

Prof. Dr-Ing. A. Koch
Jaco Hofmann, MSc.



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Wintersemester 16/17
Übungsblatt 5

In dieser Übung entwickeln Sie als Erweiterung für die, in der vorherigen Übung erstellte, RGB zu GrayScale Konvertierungseinheit einen einfachen Bildfilter.

Hinweis:

Die normale Version dieser Übung arbeitet nur auf 3×3 großen Kernen über einem 20×20 großen Bild. Möchten Sie diese Übung mit erhöhtem Schwierigkeitsgrad bearbeiten, entwickeln Sie alle folgenden Module generisch, das heißt, ihre Implementierung kann über Typparameter für beliebigen Bildgrößen und Kernelgrößen konfiguriert werden.

Aufgabe 5.1 Median

In dieser Übung wird ein 3×3 Median-Filter entwickelt. Wie der Name impliziert, muss der Median eines neunelementigen Vektors gefunden werden. Realisieren Sie ein Modul, das einen solchen Median findet. Nutzen Sie dafür eine Pipeline von Vergleichen mit jeweils drei Eingängen und Ausgängen. Die Ausgänge des eines Vergleichers sind der höchste Wert der Eingabe, der niedrigste Wert der Eingabe und der mittlere Wert der Eingabe. Pro Takt soll immer nur ein Vergleich sequentiell (beliebig viele Parallel) ausgeführt werden.

```
1 module mkMedian(Header#(Vector#(9, GrayScale), GrayScale));
2 ...
3 endmodule
```

Testen Sie Ihre Implementierung mit Hilfe von BlueCheck. Als einfache (aber ineffiziente) Vergleichsoperation können Sie die Listen-Funktion `sort` verwenden. Einen Vektor können Sie in eine Liste umwandeln mit der Funktion `toList` (BSV-Reference-Guide 259ff.).

Aufgabe 5.2 Stream Kernelbearbeiter (Schwierig)

Entwickeln Sie ein Modul, das einen Stream von GrayScale Daten als Eingabe erhält. Innerhalb des Moduls sollen 3×3 Kernel erzeugt und an das Median Modul übergeben werden. Die Ausgabe des Moduls ist das Median gefilterte Bild als Stream von GrayScale Werten. Das Eingabebild wird Zeile für Zeile von links nach rechts und oben nach unten eingelesen.

```
1 module mkMedianFilter(Header#(GrayScale, GrayScale));
2 ...
3 endmodule
```

Hinweis:

Das Bild wird durch die Bearbeitung nicht verkleinert. Beachten Sie die Ränder des Bildes nicht.

Benutzen Sie zwei `mkSizedFIFO` zum Zwischenspeichern der Bildreihen wie in Abbildung 1 skizziert. Die Pfeile geben die Datenflussrichtung an. Wenn der erste Pixel das mittlere Register (Reg) erreicht können Sie den ersten Kernel (alle Register) an das Median Modul übergeben.

Testen Sie die Module angemessen.

Erweiterung:

Behandeln Sie die Ränder des Bildes korrekt. Der Nutzer des Moduls soll den Wert der Ränder als Parameter übergeben können.

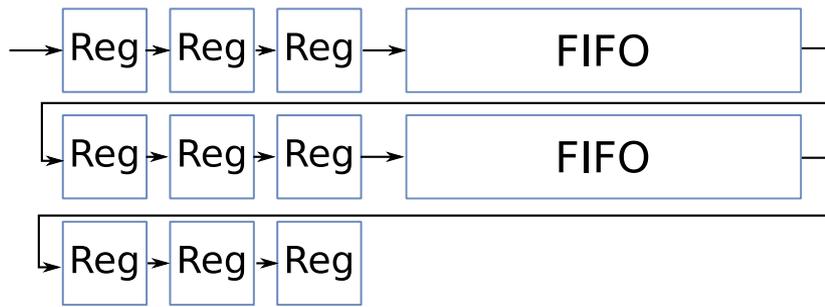


Abbildung 1: Effiziente Buffer für Bildzeilen für die Bildverarbeitung in Hardware

Aufgabe 5.3 Pipeline

Testen Sie den Farb-zu-Graustufen Konvertierer zusammen mit dem Median-Filter aus dieser Übung. Benutzen Sie `mkConnection` aus dem Paket `Connectable` für die Verbindung.