

# Praktikum 9: Parallel Sorting

M.Thaler, 2/2016, ZHAW

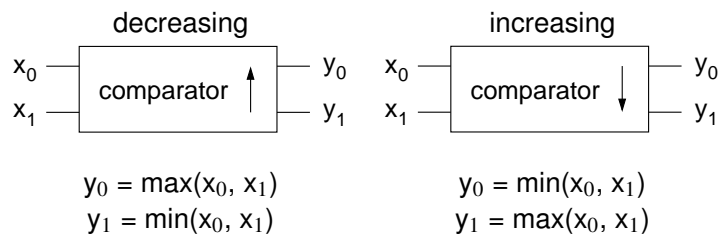
## 1 Einführung

Sortieren ist eine wichtige und häufige anzutreffende Anwendung in der Informatik. Ein Verfahren zum parallelen Sortieren auf Multicore Prozessoren haben Sie im Projekt kennengelernt. Die Frage stellt sich nun wie auf einer GPU mit hunderten von Prozessoren sortiert werden kann.

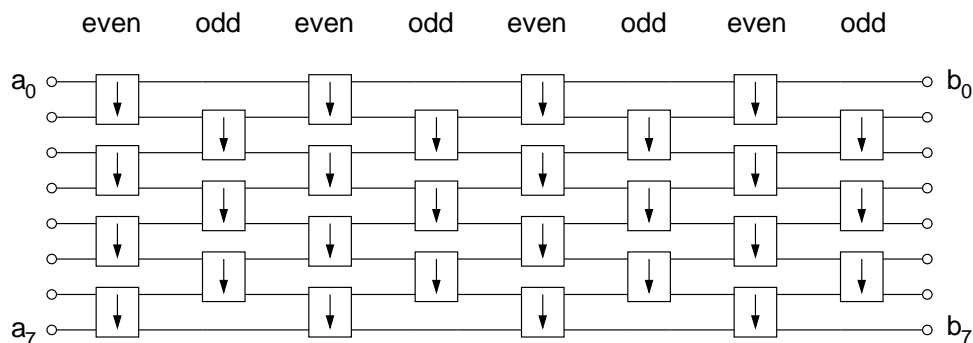
In diesem Praktikum werden Sie sogenannte Sortiernetzwerke kennenlernen, die sich speziell für Prozessoren mit sehr vielen CPUs eignen und wo die Daten noch im Speicher Platz finden.

## 2 Grundlagen

Das Grundelement bei Sortiernetzwerken besteht aus Komparatoren, die zwei Eingangswert in ab- oder aufsteigender Reihenfolge auf die Ausgänge abbilden:



Mit Hilfe dieser Element lassen sich Sortiernetzwerke aufbauen, z.B. ein even-odd Sortiernetzwerk, hier für 8 Datenwerte:



Die Komplexität dieses Algorithmus ist  $O(N^2)$ , der Rechenaufwand für  $p = N$  beträgt aber nur  $O(N)$ : es können  $O(N)$  Datenwerte gleichzeitig verarbeitet werden.

## 3 Aufgabe 1

Implementieren Sie in OpenCL einen Kernel, der eine Array (Typ unsigned int) *in place* mit Hilfe eines even-odd Verfahrens sortiert. Dabei soll die Arraylänge vorerst gleich gross, wie die Work-group gewählt werden.

Überlegen Sie sich, wie das Indexing gehandhabt werden kann und gehen Sie davon aus, dass ein lokaler Array für Zwischenwerte zur Verfügung steht. Lesen dazu auch den Hinweis auf der nächsten Seite.

Hinweis: die Implementation lässt sich vereinfachen, wenn der lokale Array um mindestens ein Eintrag grösser ist, als der Datenarray

Im Verzeichnis `parSort/a0` haben wir das Hauptprogramm und einen Rahmen für den Kernel vorbereitet. Das Hauptprogramm enthält einen einfachen Unit-Test für aufsteigendes Sortieren.

## 4 Aufgabe 2

Nun soll das Sortiervfahren auf eine beliebige Arraygrösse (Vielfaches der Workgroup Grösse) erweitert werden. Das Hauptprogramm soll dabei repetitiv Kernel aufrufen.

Hinweis: beim Einreihen eines Kernel kann ein Datenoffset angegeben werden, damit kann sowohl für die even als auch die odd Phase der gleiche Kernel verwendet werden.

## 5 Anmerkung

Ein weiteres Sortiervfahren, das in die hier vorgestellte Kategorie fällt ist, ist das sogenannte *bitonic*-Sortiervfahren, das noch etwas effizienter arbeitet (ca. Faktor 2), aber auch viel komplizierter ist.

## 6 Take Home

Notieren Sie sich die wichtigsten Punkte, die Sie im Zusammenhang mit diesem Praktikum kennengelernt haben.