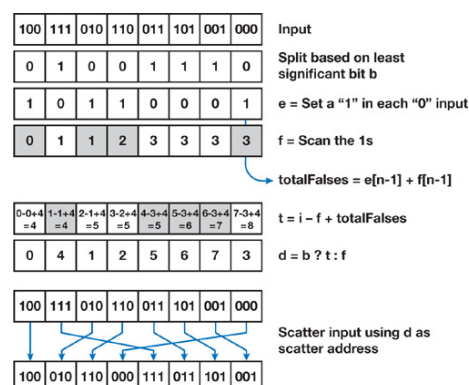


Aufgabenblatt 8

1. Aufgabe (4 + 6 Punkte)

Implementieren Sie eine parallele Version des Sortieralgorithmus Radixsort (zur Basis 2) in CUDA. Der Algorithmus soll ein `unsigned int` Array beliebiger Größe auf der Grafikkarte sortieren können.

- Implementieren Sie zunächst eine möglichst effiziente Scan-Methode, welche die exklusive Präfixsumme¹ eines Arrays beliebiger Größe auf der Grafikkarte berechnet. Verwenden Sie geteilten Speicher (Shared Memory). Orientieren Sie sich an http://http.developer.nvidia.com/GPUGems3/gpugems3_ch39.html.
- Verwenden Sie die Scan-Methode aus Teilaufgabe a) um Radixsort zu implementieren. Dazu müssen für jede Bitstelle der `unsigned int` Schlüssel folgende Schritte ausgeführt werden²:



¹<https://de.wikipedia.org/wiki/Pr%C3%A4fixsumme>

²Quelle: <https://de.wikipedia.org/wiki/Pr%C3%A4fixsumme>

Der Algorithmus kann zusätzlich in folgendem Video nachvollzogen werden: <https://www.youtube.com/watch?v=iS0S7F2U4-o>. Beachten Sie, dass ihre Radixsort-Implementierung *out-of-place* arbeiten soll und somit in mehreren Iterationen mit vertauschten Ein-/Ausgabepuffern auf dem Host aufgerufen werden muss. Vergleichen Sie die Ausführungsgeschwindigkeit ihrer Implementierung mit `std::sort` auf dem Host-Rechner.

Hinweise

- Die Abnahme für das Blatt soll bis Dienstag, 14. Juni 2016 erfolgen.

Linksammlung

- <http://hpc.oit.uci.edu/nvidia-doc/sdk-cuda-doc/>
- <http://docs.nvidia.com/cuda/cuda-c-best-practices-guide/>
- <http://docs.nvidia.com/cuda/kepler-tuning-guide/index.html>