

Prüfungsaufgabe 2

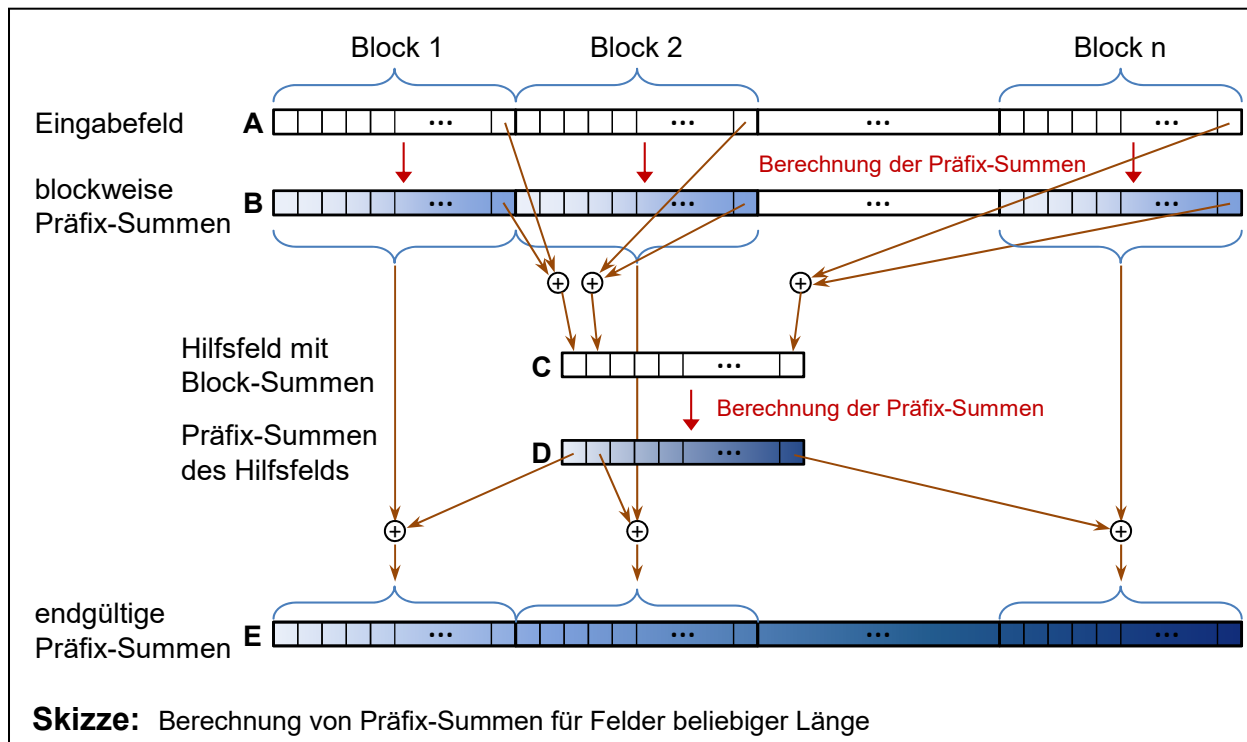
Für eine Folge von Zahlen x_0, x_1, \dots, x_n ist die Folge der Präfixsummen (exklusiv) definiert als $0, x_0, x_0 + x_1, \dots, \sum_{i=0}^{n-1} x_i$ (Summe aller vorherigen Elemente der Folge).

Beispiel:

Folge	3	2	1	2	1	4	3	2	4	3
PräfixSummen	0	3	5	6	8	9	13	16	18	22

In einer Übungsaufgabe haben Sie für Felder der Länge 256 bereits eine parallele Variante auf der GPU implementiert (im Folgenden als *Basisalgorithmus* bezeichnet).

Aufbauend auf diesem Basisalgorithmus können Präfixsummen für Felder beliebiger Länge wie folgt berechnet werden:



- 1) Falls die Feldlänge des Eingabefeldes A kein Vielfaches von 256 ist, wird das Feld bis zum nächsten Vielfachen von 256 vergrößert.
- 2) Falls Feld A die Länge 256 besitzt, werden die Präfixsummen mit Hilfe des Basisalgorithmus berechnet, andernfalls geht es bei Punkt 3) weiter.
- 3) Das Feld wird in Blöcke der Länge 256 unterteilt. Für diese Blöcke werden unabhängig voneinander mit Hilfe des Basisalgorithmus die Präfixsummen berechnet. Das Ergebnis ist ein Feld B, das die Präfixsummen der einzelnen Blöcke enthält.
- 4) In jedem Block in Feld B enthält das rechte Element nun die Summe der ersten 255 Elemente des entsprechenden Blocks in Feld A. Addiert man dazu noch das rechte Element des Blocks aus Feld A, erhält man die Summe aller 256 Elemente des Blocks in Feld A. Die Summen aller Blöcke werden nun in ein Hilfsfeld C eingetragen.
- 5) Wendet man den hier beschriebenen Algorithmus rekursiv auf Feld C an, erhält man Feld D, das die Präfixsummen des Feldes C enthält.
- 6) Feld D enthält nun für jeden Block aus Feld B den Wert, der zu allen Elementen des Blocks addiert werden muss, um das Endergebnis E (die Präfixsummen aus Feld A) zu erhalten (Das erste Element von D hat den Wert 0).

Prüfungsaufgabe 2

Aufgabe:

Implementieren Sie die Berechnung der Präfixsummen für Felder beliebiger Länge nach dem o.g. Verfahren in OpenCL. Gehen Sie dabei wie folgt vor:

- Erzeugen Sie zusätzlich zu den OpenCL-Buffern für die Felder A und B weitere Buffer für die Felder C und D.
- Erzeugen Sie die OpenCL-Buffer gleich in einer Größe, dass die auf ein Vielfaches von 256 vergrößerten Felder darin Platz finden.
- Wenn die OpenCL-Buffer in ausreichender Größe erzeugt wurden, um die vergrößerten Felder aufnehmen zu können, reicht es, nur die Elemente der unvergrößerten Felder in den vorderen Teil der Buffer zu schreiben. Die zusätzlichen Elemente können einfach uninitialized bleiben (zufälliger Wert), da deren Werte keine Rolle spielen. Wenn Sie möchten können Sie die zusätzlichen Elemente aber auch durch einen Aufruf von `clEnqueueWriteBuffer` mit entsprechendem Offset auf 0 setzen.
- Implementieren Sie eine Version des Basisalgorithmus so,
 - dass er für Felder, deren Länge ein Vielfaches von 256 ist, blockweise die Präfixsummen berechnet und Feld B als Ergebnis liefert. Die Workgroup-Größe bleibt dabei bei 256. Die `global worksize` entspricht der Feldlänge, so dass jeder 256-Elemente-Block von den Workitems einer Workgroup berechnet wird. (Beim Zugriff auf die globalen Felder A und B müssen Sie als Index die `global_id` verwenden, für den Zugriff auf das temporäre Feld im lokalen Memory die `local_id`.)
 - dass er Feld C gleich mit befüllt. Nach dem Kopieren der Präfixsummen aus dem lokalen ins globale Memory (Feld B) kann z.B. das Workitem, das das rechte Element schreibt, anschließend die Summe aus diesem Element und dem rechten Element des Blockes aus Feld A in Feld C eintragen. Den Index für den Eintrag in Feld C können Sie mit `get_group_id(0)` bestimmen.
- Gegen Abzug von 1/3 Note können Sie auf die rekursive Berechnung der Präfixsummen von Feld C verzichten und stattdessen unter der Annahme, dass C auf 256 Elemente beschränkt ist, einfach den Basisalgorithmus anwenden. Ihr Algorithmus ist dann auf Felder der Länge 65536 begrenzt.
- Wenn Sie sich für die rekursive Variante entscheiden, ist es sinnvoll für den Gesamtalgorithmus eine rekursive Funktion zu verwenden, die die OpenCL-Buffer für die Felder A und B bereits als Parameter beim Aufruf geliefert bekommt. Der OpenCL-Buffer für Feld A soll bereits vor dem Aufruf der rekursiven Funktion mit den Eingabedaten befüllt werden, das Ergebnis wird im OpenCL-Buffer für B geliefert. Innerhalb der Funktion erzeugen Sie die OpenCL-Buffer für die Felder C und D. Diese können dann beim Rekursiv-Aufruf übergeben werden. Dadurch ist es möglich, die Zwischenergebnisse in den Feldern C und D ausschließlich in den OpenCL-Buffern zu halten und ein Hin- und Her-Kopieren zwischen OpenCL-Buffern und Main-Memory zu vermeiden.
- Zur Berechnung des endgültigen Ergebnisses (Feld E) benötigen Sie einen weiteren Kernel mit den OpenCL-Buffern der Felder B und D als Eingabe. Benutzen Sie auch hierfür eine Workgroup-Größe von 256 und berechnen Sie in jedem Workitem ein Element. Für das Endergebnis (Feld E) ist kein weiterer OpenCL-Buffer nötig. Die Einträge in Feld B können durch das Ergebnis überschrieben werden.