christian.hakert [☺] tu-dortmund.de
marcel.ebbrecht [☺] tu-dortmund.de

Übung zur Vorlesung
Rechnerarchitektur
Sommersemester 2020

# Übungsblatt 6

**Bearbeitung ab Montag, 15. Juni 2020**

## 6.1 OpenCL - Theorie

Betrachten Sie folgende Code-Fragmente (C Version / C++ Version):

| | |
|---|---|
| 1 | **memoryObjects[0] = clCreateBuffer(context,CL_MEM_READ_ONLY \| CL_MEM_ALLOC_HOST_PTR, buffSize, NULL, &err);**<br>**cl::Buffer buffer_0(CL_MEM_READ_ONLY \| CL_MEM_ALLOC_HOST_PTR, buffSize);** |
| 2 | **cl_int* inputA = (cl_int*)clEnqueueMapBuffer(commandQueue, memoryObjects[0],CL_TRUE, CL_MAP_WRITE, 0, bufferSize, 0, NULL, NULL, &errorNumber);**<br>**cl_int* inputA = (cl_int*)cl::commandQueue::enqueueMapBuffer(buffer_0,CL_TRUE, CL_MAP_WRITE, 0, bufferSize);** |
| 3 | **int i = get_global_id(0);**<br>**int i = get_global_id(0);** |

Bitte erklären Sie, was das jeweilige Code-Fragment bedeutet bzw. macht.

## 6.2 OpenCL - Eigenständiges Erweitern

Führen Sie folgende Änderungen am 'Hello_World_OpenCL'-Programm, mit welchem Sie sich bereits letzte Woche ausführlich beschäftigt haben, durch:

(a) Erweitern Sie das 'Hello_World_OpenCL'-Programm so, dass statt zwei Inputwerten und einem Outputwert drei Inputwerte und zwei Outputwerte erzeugt werden (im Folgenden benannt als W1, W2, W3, O1 und O2).

(b) Ändern Sie das Programm nun so, dass mit cl_float anstatt mit cl_int gerechnet wird.

(c) Programmieren Sie das Programm so um, dass zwei unterschiedliche Kernel nacheinander ausgeführt werden:

(i) Der erste Kernel soll (W1+W2+W3)*(W1-W2-W3) berechnen und in O1 speichern.

(ii) Der zweite Kernel soll die Nachkommastellen der Zahl mit der Zahl vor dem Komma für jeden der drei Inputs multiplizieren und die Summe dieser Ergebnisse in O2 speichern. Für 337,4284 wäre das z.B. 337*0,4284=144,3708.

## 6.3 OpenCL - Zusatzaufgabe

Führen Sie folgende Änderungen am 'Hello_World_OpenCL'-Programm, mit welchem Sie sich bereits letzte Woche ausführlich beschäftigt haben, durch:

(a) Legen Sie ein cl_int-Array an.

(b) Füllen Sie das Array mit Zufallswerten zwischen 0 und 255.

(c) Erstellen Sie einen Kernel, welcher jedes Feld neu berechnet. Dabei soll jeder Wert so angepasst werden, dass er 50% näher am Durchschnittswert der umliegenden Felder ist. Umliegende Felder wären bei Feld 1 zum Beispiel die Felder 2,4 und 5, deren Durchschnitt dann berechnet werden muss. Danach benötigt man noch die Differenz zwischen Ursprungswert und Durchschnitt, welche halbiert werden muss (50%) und dann auf den Ursprungswert addiert wird, um den neuen Wert zu erhalten. (siehe Beispiel)

**Beispiel:**

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

In diesem Fall wären die neuen Werte für die Felder:

| | | |
|---|---|---|
| Feld 1: | 1 + (((2+4+5)/3-1)/2) | =2 |
| Feld 2: | 2 + (((1+3+4+5+6)/5-2)/2) | =2 |
| Feld 3: | 3 + (((2+5+6)/3-3)/2) | =3 |
| Feld 4: | 4 + (((1+2+5+7+8)/5-4)/2) | =4 |
| Feld 5: | 5 + (((1+2+3+4+6+7+8+9)/8-5)/2) | =5 |
| Feld 6: | 6 + (((2+3+5+8+9)/5-6)/2) | =6 |
| Feld 7: | 7 + (((4+5+8)/3-7)/2) | =6 |
| Feld 8: | 8 + (((4+5+6+7+9)/5-8)/2) | =7 |
| Feld 9: | 9 + (((5+6+8)/3-9)/2) | =8 |

**Allgemeine Hinweise:** Informationen zur Veranstaltung finden Sie unter hhttps://ls12-www.cs.tu-dortmund.de/daes/de/lehre/lehrveranstaltungen/summersemester-2020/rechnerarchitektur-deutsch.html. Eine Abgabe der Übungsblätter ist nicht erforderlich.