第5章 ホスト-カーネル間のデータ受け渡し／マルチスレッド処理／カーネルの制約事項

動かしながら学ぶ！OpenCLコーディング

本橋 弘臣

本稿からは，第4章で作成したプログラムchap4/opencl-hello-error-checkをベースに，機能を追加しながらGPUプログラミングを学んでいきます．opencl-hello-error-checkからの変更箇所には，//☆の目印を追加してあります．

本稿では，基本となる次の3つについて説明します．

- ホスト-カーネル間のデータ受け渡し
- マルチスレッド処理
- OpenCLカーネルの制約事項

1 ホスト-カーネル間のデータ受け渡し

カーネル関数に引数を渡す方法

chap5-1/opencl-arg-test

● カーネル関数

カーネル関数に引数を渡す処理を追加します．**リスト1**(a)に示すカーネル関数は，引数としてint型のaとbを受け取り，c = a + bの計算を行った結果を表示しています．

リスト1　ホスト・コードからカーネル関数に引数を受け渡すコード(chap5-1/opencl-arg-test)

```
__kernel void arg_test(int a, int b)
{
  int c;

  c = a + b;
  printf("[Kernel] %d + %d = %d\n", a, b, c);
}
```

(a) カーネル関数

```
cl_int  ha = 10, hb = 20;

// カーネルの引数を設定する
clSetKernelArg(kernel, 0, sizeof(cl_int),
                                    (void *)&ha);
clSetKernelArg(kernel, 1, sizeof(cl_int),
                                    (void *)&hb);

// カーネルを実行する
clEnqueueNDRangeKernel(queue, kernel, 1, NULL, gws,
                              lws, 0, NULL, NULL);
```

(b) ホスト・コード側の追加コード

● ホスト・コード側…カーネル実行前に引数の値を設定

このカーネル・コードを呼び出すホスト・コード側には，**リスト1**(b)の処理を追加しています．カーネルを実行する前にclSetKernelArg()関数を呼び出すことにより，カーネル関数に渡したい引数の値を設定します．引数が1個増えるたびにAPI呼び出しが1回増えてしまうので，若干煩雑な感がありますが，引数を値で渡す場合にはこのようなコードになります．この方法とは別に，たくさんのデータをポインタ渡しする方法もあるのですが，その方法は後ほど説明します．

clSetKernelArg()関数の第2引数は，何個目の引数なのかを示すパラメータです．またカーネルに渡したい値を入れる第4引数にはポインタを渡す必要があるので，カーネルに何か値を渡したい場合には，サンプル・コードのようにその値をあらかじめ変数に格納しておく必要があります．

● 実行結果

このプログラムの実行結果を次に示します．

```
$ ./main
 :
[Kernel] 10 + 20 = 30
```

カーネル関数の引数にclSetKernelArg()で設定した変数の値が正しく渡されています．