

به نام خدا



برنامه‌نویسی چند هسته‌ای

دستور کار آزمایشگاه ۷

هدف آزمایش:

در این آزمایش با استفاده از ابزار پروفایلینگ به اندازه‌گیری میزان Occupancy در یک مثال ساده پرداخته و نشان می‌دهیم افزایش میزان Occupancy لزوماً به معنی افزایش سرعت کد نیست. همچنین نحوه استفاده از استریم برای هم‌زمان کردن اجرای کارها روی gpu نشان داده خواهد شد

```
#define MAX_HISTOGRAM_NUMBER 10000
```

```
#define ARRAY_SIZE 102400000
```

```
#define SCALER 80
```

```
#define CHUNK_SIZE 100
```

پارامتر اول برای مشخص کردن بازه اعداد موردبررسی و پارامتر دوم تعداد آنها را مشخص میکند. با استفاده از پارامتر `scaler` میزان کار هر نخ با انجام محاسبات تکراری افزایش می‌یابد

عدد پیشنهادی برای `Scaler` در GPU با معماری پاسکال ۸۰ است

ترجیحاً `Scaler` را برای هر قسمت آزمایش ثابت قرار دهید.

قسمت اول: بررسی اثر occupancy بر میزان تسریع

این تابع کرنل و همراه کد `host` در فایل `histogram.cu` موجود است. ابتدا کدها را مطالعه کنید و پس از اطمینان از درک درست آزمایش با توجه به اندازه آرایه ورودی و تغییر `THREAD_COUNT` و `CHUNK_SIZE` تعداد بلوکهای مورد نیاز برای فراخوانی تابع را بدست می‌آورید و سپس جدول زیر را کامل کنید.

```

__global__ void histogramKernelSingle(unsigned long long int *c, int *a)
{
    unsigned long long int worker = blockIdx.x*blockdim.x + threadIdx.x;
    unsigned long long int start = worker * CHUNK_SIZE;
    unsigned long long int end = start + CHUNK_SIZE;
    for (int ex = 0; ex < SCALER; ex++)
        for (long long int i = start; i < end; i++)
        {
            if (i < ARRAY_SIZE)
                atomicAdd(&c[a[i]], 1);
            else
            {
                break;
            }
        }
}

```

اندازه‌ی آرایه ورودی و بازه اعداد داده شده در کد در بالا آمده است .

Thread count	<u>8</u>	<u>16</u>	<u>32</u>	<u>256</u>	<u>1024</u>
Scaler					
Number of Blocks					
Theoretical Occupancy					
Achieved Occupancy					
Kernel Execution Time					

دلیل کاهش سرعت با افزایش Occupancy چه بوده است؟

قسمت دوم : استفاده از استریم

کد قسمت قبل را با تغییرات مناسب جهت استفاده از Stream تغییر دهید و کرنل مربوطه را ۴ بار فراخوانی کنید هربار Scaler را برابر ۲۰ قرار دهید و سپس جدول زیر را کامل کنید.

Thread count	8	16	32	256	1024	8 ^h	16	32	256	1024
Scaler										
Chunk size										
Stream										
Number of blocks										
Theoretical Occupancy										
Fully Concurrent Streams										
Achieved Occupancy										
Kernel Execution Time										

برای مشاهده اجرای همزمان Stream ها باید از Visual profiler استفاده نمایید

پس از انجام قسمت های بالا سوالات زیر را جواب دهید :

۱. به نظر شما Stream ها برای چه نوع مساله‌هایی مناسب تر است؟
۲. آیا با زیاد کردن Stream میزان Occupancy تغییر می کند؟
۳. کاهش یا افزایش تعداد بسته‌های هیستوگرام چه تاثیری روی occupancy دارد