# 需求

这个项目基本上是把现在的再细化一下，把结构都弄清楚，然后写一篇完整的测试报告。

1.不同形式的数据组织形式

2.不同的访问方式

3.不同的内存

4.不同的线程数目

# 需求理解

这个实验需要考虑的因素包括：**数据存储方式（**一维数组、二维数组等**），数据拷贝方式（**直接拷贝、对齐拷贝、数组拷贝等**），数据存储位置（**全局内存、共享内存、纹理内存、常量内存**），数据访问方式（**顺序访问、step访问、访问次数**）、**以及共享内存的bank conflict问题、纹理内存的越界访问优势、常量内存的带宽限制等等

我认为，**这个Case是由需求驱动的。不同的情景下，最佳的访问方式可能截然不同。实验设计者需要对当前CUDA内常用的访问方式比较了解，才能较有意义地设计这个Case。**

# 实验设计

**不同形式的数据组织形式**：

一维数组。

二维数组等长数组。

**数据拷贝方式**：直接拷贝（cudaMalloc,cudaMemcpy）另：对齐拷贝对比（cudaMallocPitch,cudaMemcpy2D）适用于一维模拟二维等长数组，这里暂时不做考虑。

**数据存储位置**：全局内存和共享内存。

**数据访问方式**：顺序访问和step访问（step=2,5）

**注**：

（1）测试数据大小受共享内存大小限制。

（2）为了防止编译器优化代码的影响，核函数内做赋值操作。

# 实验环境

硬件环境包括：

|  |  |
| --- | --- |
| 项目 | 描述 |
| CPU | Intel Core i7 2600 |
| 内存 | 12GB DDR3 1333MHz |
| 硬盘 | WD 绿盘 1TB |
| 显卡 | GeForce GTX TITAN |

软件环境包括：

|  |  |
| --- | --- |
| 项目 | 描述 |
| 操作系统1 | Ubuntu 14.04 Desktop x64 |
| CUDA Version | 7.0 |

# 实验结果

见statistics.txt