

```

#include <stdio.h>
#include <cuda.h>
#include <cuos.h>

#define N 100
#define M 32
#define K 2

__device__ volatile int vint = 0;

1. __global__ void
entry( volatile int* foo )
{
    for (int i = 0; i < N; ++i) {
        atomicAdd((int*)foo, 1);
    }
}

extern "C"
__global__ void
diverge_cta( volatile int *foo )
{
    __shared__ int x;
    if ((threadIdx.x%32) != 0) {
        return;
    }
    if (threadIdx.x == 0) {
2.     entry<<<1,M>>>( foo );
3.     cudaDeviceSynchronize();
        x = 5;
        return;
    }
    __syncthreads();

    atomicAdd((int*)foo, x);
}

```