ECE 408 Exam 1, Fall 2024
Tuesday, October 15, 7:00 – 9:00 PM

Name: _____

NetID: _____

UIN: _____

- **Be sure your exam booklet has exactly XX pages.**
- **Write your name at the top of each page.**
- **Do not tear the exam apart.**
- **This is a closed-book exam; NO handwritten notes are permitted.**
- **You may not use any electronic devices except for a simple calculator.**
- **Absolutely no interaction between students is allowed.**
- **Show all work, and clearly indicate any assumptions that you make.**
- **Illegible answers will likely be graded as incorrect.**
- **Don't panic, and good luck!**

Problem X	XX	points	_____
Problem X	XX	points	_____
Problem X	XX	points	_____
Total	100	points	_____

Reference Sheet

Execution Space Specifier

`__global__` declares a function as being a kernel. A `__global__` function must have `void` return type, and cannot be a member of a class. A call to a `__global__` function is asynchronous, meaning it returns before the device has completed its execution.

`__host__` declares a function that is (a) executed on the host and (b) callable from the host only.

`__device__` declares a function that is (a) executed on the device and (b) callable from the device only.

Variable Memory Space Specifier

`__constant__` declares a variable that resides in constant memory space.

`__shared__` declares a variable that resides in the shared memory space of a thread block.

Synchronization

`void __syncthreads ( )`
Wait until all threads in the thread block have reached this point and all global and shared memory accesses made by these threads prior to `__syncthreads` are visible to all threads in the block.

`cudaError_t cudaDeviceSynchronize ( void )`
Wait for compute device to finish.

Built-in Variables

`gridDim` is of type `dim3` and contains the dimensions of the grid. Access the number of blocks in the grid in each dimension using `gridDim.x`, `gridDim.y`, and `gridDim.z`.

`blockDim` is of type `dim3` and contains the dimensions of the block. Access the number of threads in the block in each dimension using `blockDim.x`, `blockDim.y`, and `blockDim.z`.

`threadIdx` is of type `uint3` and contains the thread index within the block. Access the thread index in each dimension using `threadIdx.x`, `threadIdx.y`, and `threadIdx.z`.

Kernel Configuration and Kernel launch

```
dim3 dimGrid(x_size, y_size, z_size);
dim3 dimBlock(x_size, y_size, z_size);
```

When defining a variable of type `dim3`, any component left unspecified is initialized to 1.

`kernelName<<<dimGrid, dimBlock>>> (input parameters)`

The number of threads per block and the number of blocks per grid specified in the `<<<...>>>` syntax can be of type `int` or `dim3`.

Memory Management

`cudaError_t cudaMalloc ( void** devPtr, size_t size )`
Allocates `size` bytes of linear memory on the device and returns in `*devPtr` a pointer to the allocated memory.

```
cudaError_t cudaMemcpy ( void* dst, const void* src,
                         size_t count, cudaMemcpyKind kind )
```
Copies `count` bytes from the memory area pointed to by `src` to the memory area pointed to by `dst`, where kind specifies the direction of the copy, and must be one of `cudaMemcpyHostToHost`, `cudaMemcpyHostToDevice`, `cudaMemcpyDeviceToHost`, or `cudaMemcpyDeviceToDevice`.

```
cudaError_t cudaMemcpyToSymbol ( const void* symbol, const void* src,
                                 size_t count, size_t offset = 0,
                                 cudaMemcpyKind kind = cudaMemcpyHostToDevice )
```
Copies `count` bytes from the memory area pointed to by `src` to the memory area pointed to by `offset` bytes from the start of symbol `symbol`. The memory areas may not overlap. `symbol` is a variable that resides in global or constant memory space. `kind` can be either `cudaMemcpyHostToDevice` or `cudaMemcpyDeviceToDevice`.

`cudaError_t cudaFree ( void* devPtr )`
Frees the memory space pointed to by `devPtr` on the device.

Profilling Tools

compute-sanitizer is a suite of tools used for checking the correctness of the kernel. The *memcheck* tool can be used for detecting and attributing out-of-bounds and misaligned memory access errors. The *inicheck* tool is used for checking uninitialized access to global memory. The *racecheck* can be used for detecting data race in shared memory access. The *synccheck* can be used for detecting invalid synchronization primitive calls.

Nsight Compute (ncu) is a suite of tools that provide detailed performance metrics and API debugging for CUDA kernels. The *interactive debugger* can be used to step through CUDA calls. The *ncu-rep report* collects key performance metrics regarding computation, memory, stalls, warp states, and occupancy of kernels.

Nsight System (nsys) is a suite of tools that perform system-wide performance analysis interactively or automatically. The *nsys-rep* report visualizes GPU and CPU activity data on a unified timeline.

Miscellaneous Functions

`ceil(x)` returns the smallest integer value greater than or equal to `x` (as a floating-point value).

`floor(x)` returns the largest integer value less than or equal to `x` (as a floating-point value).

`sqrtf(x)` returns the square root of the value `x`, where `x` is a non-negative floating-point number. The result is also a floating-point value.

`sizeof(type)` yields the size of `type` in bytes.