ECE 786 Assignment 1 Report

Part A:

Question: explain your CUDA kernel function and report the timing differences with different memory management method.

Solution:

```
_global__ void quamsim (const float *U_matrix,const float *input_matrix, float *output_matrix,int
num_of_elements, int q_bit)
{
      int threadid = blockDim.x*blockIdx.x + threadIdx.x;
      int q_id = threadid^(1<<q_bit);

      if (threadid < num_of_elements)
      {
      if ((threadid& 1<< q_bit) ==0){
            output_matrix[threadid] = (U_matrix[0]*input_matrix[threadid]) +
(U_matrix[1]*input_matrix[q_id]);
            output_matrix[q_id] = (U_matrix[2]*input_matrix[threadid]) +
(U_matrix[3]*input_matrix[q_id]);
       }
      }
}
```

Above is the kernel function U_matrix is the 2X2 matrix, input_matrix is the input_matrix given by the given file, output_matrix is the output which needs to be printed, q_bit is the position which is taken from the Input file.
q_id is the intermediate value which calculated by taking the bitwise Xor after 1 is shifted left q_bit amount of time , to isolate the position.
Num_of_elements is the total number of elements in the given input file.
Now we need to check the condition where theardid is anded after left shift of 1 with q_bit number of positions and the output value is zero then the value of the output is calculated.

Now the time calculated for the first version that is once the values are copied from the host memory to the device memory and again the ouput is copied back using cudaMemcpy() is 300 us for 128 bit long vector.
On the contrary the time take with cudamemorymanaged()(i.e. using unified virtual memory) is 330 us.

I conclusion we know that the UVM method takes more time than the first approach.

It second approach is easy to code as compared to the first approach , but it looks like the first approach is more efficient than the second approach.

Part B:
Question:

a)What is the IPC of your program and how is this value calculated from the statistics?

Solution:

No. of cycles = 5595

Number of instructions = 5824

Instruction Per Cycle = 1.0409

Tnstruction per cycle is caluated by Total Number of instructions/Total number of cycles

b) What is the data cache miss_rate and how is this value calculated from the statistics?

Solution:

L1 total_cache_access = 64

L1 total_cache_missess = 40

L1 total_cache_miss_rate = 0.625

Cache miss rate is calculated by taking the ratio of L1_cache_miss/ L1_total_cache_access

L2_total_cache_access = 33

L2_cache_miss = 16

L2_miss_rate = 0.4848

Cache miss rate is calculated by taking the ratio of L2_cache_miss/ L2_total_cache_access