Assignment 11

The due date for submitting this assignment has passed. **Due on 2020-04-22, 23:59 IST.** As per our records you have not submitted this assignment.

1) Consider the following kernel code snippet.

```c
#define WarpSize 32
__global__ void assign_lane ( float * g_data)
{
    int lid = threadIdx.x;
    int gid = blockIdx.x*blockDim.x + threadIdx.x;
    int lane = lid&WarpSize;
    g_data[gid]=lane;
}
```

The above kernel is launched with the launch parameters `<<<16,128>>>.` After the values of `g_data` are assigned, a standard reduction kernel is called which calculates the sum of all the elements in `g_data`. What is the final sum?

- a) 31744
- b) 32768
- c) 16536
- d) 65536

b)
2) Consider the following code snippet executing on a GPU architecture where the number of shared memory banks is 8 and the bank width is 4 bytes.

```c
#define SZ 32
__global__ void setRowCol(float *out) {
    __shared__ int tile[SZ][SZ];
    unsigned int gid = threadIdx.x * blockDim.y + threadIdx.y;
    tile[threadIdx.x][threadIdx.y] = gid;
    __syncthreads();
    int tidx = threadIdx.x;
    int tidy = threadIdx.y;
    out[gid] = tile[tidx][tidy];
}
```

Assume that warp size for the architecture is 8. The kernel is launched with the parameters `<<<(1,2),(8,8)>>=`

Which of the following is true?

a) shared memory bank conflicts occur for shared reads and loads

b) shared memory bank conflicts do not occur for shared reads and loads

No, the answer is incorrect.
Score: 0
Accepted Answers:

a)

3) 5 points

COMMON DATA FOR QUESTIONS 3 TO 6
Fill in the blanks for the given OpenCL code snippet for coalesced version matrix vector multiplication.

```c
__kernel void MatVecMulCoalesced0(__global float * M, __global float * V, int width, int height, __global float * W )
{
    __local float partialDotProduct[2048];
    for (int y = get_group_id(0); y < height;
        y += get_num_groups(0))
    {
        __global float *row = __attribute__((global))
            float sum = 0;
        for (int x = get_local_id(0); x < width; x +=
            get_local_size(0))
            sum += row[x] * V[x];
        partialDotProduct[__attribute__((global))]
            = sum;
        C___;
        if (get_local_id(0) == 0)
        {
            float dotProduct = 0;
            for (int t = 0; t < get_local_size(0); ++t)
                W[y] = dotProduct;
        }
    }
    barrier(CLK_LOCAL_MEM_FENCE);
}
```

**Options for A:**

a) M + x * width  
b) row + y * width  
c) M + y * width  
d) V + x * width

- a)  
- b)  
- c)  
- d)  

No, the answer is incorrect.  
Score: 0  
Accepted Answers:  
c)

4) **Options for B:**

```c
    a) get_local_id(0)  
    b) get_local_size(0)  
    c) get_global_id(0)  
    d) get_local_id(1)
```

- a)
No, the answer is incorrect.
Score: 0
Accepted Answers:
a)

5) **Options for C:**

a) Nothing
b) barrier(CLK_LOCAL_MEM_FENCE);
c) W[y] = sum;
d) __syncthreads();

No, the answer is incorrect.
Score: 0
Accepted Answers:
b)

6) **Options for D:**

a) dotProduct *= partialDotProduct[width];
b) dotProduct *= partialDotProduct[t];
c) dotProduct *= partialDotProduct[t/2];
d) dotProduct += partialDotProduct[t];

No, the answer is incorrect.
Score: 0
Accepted Answers:
d)

**COMMON DATA FOR QUESTIONS 7 & 8**

7)  

**10 points**
Consider the two given scenarios and choose which type of fusion is more efficient for a GPU which has 16 SMs and can accommodate 2048 threads at max per SM. Also calculate the total number of cycles it takes for the fused kernel to execute on the given architecture.

Scenario 1

Launch parameters (CUDA) and number of cycles to complete for -
K1: <<< 8, 2048 >>>, 30 cycles
K2: <<< 8, 2048 >>>, 40 cycles
K3: <<< 16, 2048 >>>, 70 cycles

Options

a) Inner thread fusion, 140 cycles
b) Inner thread fusion, 100 cycles
c) Inter block fusion, 110 cycles
d) Inter block fusion, 100 cycles

No, the answer is incorrect.
Score: 0
Accepted Answers:
c)

8)
Scenario 2

Launch parameters (CUDA) and number of cycles to complete for -
K4: <<< 16, 1024 >>>, 40 cycles
K5: <<< 16, 1024 >>>, 40 cycles

Options

a) Inner thread fusion, 80 cycles
b) Inner block fusion, 80 cycles
c) Inter block fusion, 80 cycles
d) Options a and b
e) All of them

No, the answer is incorrect.
Score: 0
Accepted Answers:
d)
COMMON DATA FOR QUESTIONS 9 TO 13

9) 4 points
Fill in the CUDA code snippet for block level coarsened version (coarsening factor is 2) of a kernel to find the sum of all elements of a vector.

```c
__global__ void block_coarsened_sum ( float * g_idata , float * g_odata , unsigned int n)
{
    int tid = threadIdx.x;
    int i0 = 2* blockIdx.x* blockDim.x + threadIdx.x;
    int il = ______________A____________;

    __shared__ float sdata0 [ BLOCK_SIZE ];
    __shared__ float sdata1 [ BLOCK_SIZE ];
    sdata0 [tid ] = (i0 < n) ? g_idata [i0] : 0;
    sdata1 [tid ] = (il < n) ? g_idata [il] : 0;
    __syncthreads () ;

    for ( unsigned int s = blockDim.x /2; s >0; s>>=1) 
    {
        if ( tid < s )
        {
            ______________B______________;
            ______________C______________;
        }
        __syncthreads () ;
    }
    if ( tid == 0 )
    {
        ______________D______________;
        ______________E______________;
    }
}
```

Options for A:

a) (2* blockIdx.x) * blockDim.x + threadIdx.x
b) (2* blockIdx.x + 1) * blockDim.x + threadIdx.x
c) ( blockIdx.x+1) * blockDim.x + threadIdx.x
d) ( blockIdx.x+1 + 1) * blockDim.x + threadIdx.x

- a)
- b)
- c)
- d)

No, the answer is incorrect.
Score: 0
Accepted Answers: 
b)
10) **Options for B:**

   a) `sdata0 [tid ] += sdata0 [ tid + s]`
   b) `sdata1 [tid ] += sdata1 [ tid + s];`
   c) `sdata0 [tid + 1] += sdata0 [ tid + s]`
   d) `sdata1 [tid + 1] += sdata1 [ tid + s];`

   ○ a)
   ○ b)
   ○ c)
   ○ d)

No, the answer is incorrect.
Score: 0
Accepted Answers:  
a)

11) **Options for C:**

   a) `sdata0 [tid ] += sdata0 [ tid + s]`
   b) `sdata1 [tid ] += sdata1 [ tid + s];`
   c) `sdata0 [tid + 1] += sdata0 [ tid + s]`
   d) `sdata1 [tid + 1] += sdata1 [ tid + s];`

   ○ a)
   ○ b)
   ○ c)
   ○ d)

No, the answer is incorrect.
Score: 0
Accepted Answers:  
b)

12) **Options for D:**

   a) `g_odata [blockIdx .x] = sdata0 [0];`
   b) `g_odata [blockIdx .x] = sdata1 [0];`
   c) `g_odata [2* blockIdx .x] = sdata0 [0];`
   d) `g_odata [2* blockIdx .x +1] = sdata1 [0];`

   ○ a)
   ○ b)
   ○ c)
   ○ d)

No, the answer is incorrect.
Score: 0
Accepted Answers:  
c)
13) **Options for E:**
   a) \( \text{g\_odata}[\text{blockIdx} \cdot x] = \text{sdata0}[0] \);
   b) \( \text{g\_odata}[\text{blockIdx} \cdot x] = \text{sdata1}[0] \);
   c) \( \text{g\_odata}[2 \cdot \text{blockIdx} \cdot x] = \text{sdata0}[0] \);
   d) \( \text{g\_odata}[2 \cdot \text{blockIdx} \cdot x + 1] = \text{sdata1}[0] \);

   No, the answer is incorrect.
   Score: 0
   Accepted Answers:
   d)