CS2323 Computer Architecture 2019

Homework 4

Important: Your submission should be named as RollNumber_CA_HW4.pdf. For example, if your roll number is cs16mtech11075, then your submission should be cs16mtech11075_CA_HW4.pdf. Except pdf, no other format is acceptable. **10 marks will be deducted for not following these instructions or if you submit a zipped file.**

To reduce TA efforts, please answer the questions in the order in which they are given. There is a bonus of 2 marks for writing your answers in microsoft word or latex or any text-editing software. This is to discourage hand-written submissions.

1. [2 marks] What is the general formula(in terms of n, where n is dimension of nxn matrix) for the time steps in which an array multiplication could be done using systolic array. Do not count T=0, since there is no computation at T=0.
2. [2 marks] The following is the code used for convolution. Rewrite the following code after tiling the first loop below.

```
for ( row=0; row<R; row++)

 for ( col =0; col<C; col++)

  for ( to=0; to<M; to++)

   for ( ti =0; ti <N; ti++)

 for ( i =0; i<K; i++)

  for ( j =0; j<K; j++)

   Output_fmaps [to] [row] [col] += Weights [to] [ti] [i] [j] * Input_fmaps[ti] [S*row+i] [S*col+j]
```

3. [1 mark] In this code, put correct qualifier in front of each function, i.e., replace __??__ with the suitable qualifier. Do not assume any default qualifier. In your answer, you need to just write the header of a function with correct qualifier.

```
#define N 16
```

```
__??__ void addFunc1(int *a, int *b, int *c) {
 *c = *a + *b; }

__??__ void addFunc2(int *a, int *b, int *c) {
 for(int i=0; i< N; i++)
   addFunc1(a+i, b+i, c+i);  }

__??__ void random_ints(int* x, int size) {
            for (int i=0;i<size;i++)
            x[i]=rand()%50;            }

__??__ int main(void) {
int *a, *b, *c;  int *d_a, *d_b, *d_c;  int tempCounter =0; int size = N * sizeof(int);
cudaMalloc((void **)&d_a, size); cudaMalloc((void **)&d_b, size); cudaMalloc((void **)&d_c, size);
a = (int *)malloc(size); random_ints(a, N); b = (int *)malloc(size); random_ints(b, N);
c = (int *)malloc(size); cudaMemcpy(d_a, a, size, cudaMemcpyHostToDevice);
cudaMemcpy(d_b, b, size, cudaMemcpyHostToDevice);
addFunc2<<<1,1>>>(d_a, d_b, d_c);
cudaMemcpy(c, d_c, size, cudaMemcpyDeviceToHost);
            free(a); free(b); free(c); cudaFree(d_a); cudaFree(d_b); cudaFree(d_c); }
```

4. [3 marks] Assume a GPU SM (streaming multiprocessor) has 64 registers. Each register is 32b. The following code executes on GPU. Write in which location will the variable/array (shown in table below), be stored.

```
__device__ int maxValue;
```

```
__something__ void render(char *ABC, int width, int height){
  unsigned int x_dim = blockIdx.x*blockDim.x + threadIdx.x;
  unsigned int y_dim = blockIdx.y*blockDim.y + threadIdx.y;
 int iteration = 0;   int pqr[4];
  //some code which modifies iteration
  if (iteration == 256)
   ABC[x_dim] = 10;
}
```

Variables	Location
x_dim	
y_dim	
iteration	
pqr	
ABC	
maxValue	

5. [2 marks] Consider the code uploaded with L17 for cache tiling in matrix-transpose. Let arrLen = 16, tiling = 4 and block size =4, cacheSize =32.
   (a) What is the dimension of matrix (in K*K) and size of cache in bytes.
   (b) Find total hits/misses for unblocked and blocked cache. For each of these, show how many misses are coming from the input and output matrix.

6. [3 marks] Consider multiplication of matrices A*B, where each matrix is 2*2. We use systolic arrays for matrix multiplication. Show alignment in time of their elements and snapshot from T=0 to T=4. You can draw with hand and then scan.

7. [2 marks] Consider the 32b value Z=01000000101101100110011001100110, which is single-precision IEEE-754 representation of the decimal number 5.7, as can be seen from https://www.h-schmidt.net/FloatConverter/IEEE754.html website. Consider the following 4 strategies: (A) fetching left-most 9 bits (B) fetching left-most 12 bits (C) fetching left-most 15 bits (D) fetching left-most 18 bits. (The bits which are not fetched are assumed to be zero).

   For each of the 4 strategies, find the actual (approximate) value of Z we obtain after fetching. Write your answer as a decimal number. (This question is from a previous lecture which will not be covered in Exam2).

8. [2 mark] Assume that SimpleRISC has two vector registers, with a vector-width of 2. We want to load contiguous data, starting at memory address r2+20 to the vector register vr1. Write a single-line instruction to achieve this. Also explain the semantics (as shown in the slides).
9. [4 marks] A sparse matrix is one where many elements are zero. Consider

```
for(i=0; i<N; i++)
 for(j=0; j<N; j++)
  X[i][j] = Y[i][j]+Z[i][j];
```

   We have two cases: (1) when input matrices are dense and (2) when matrices are sparse such that, in each input matrix, only N*N/4 elements are non-zero at the same location. (Side point: Dense matrices arise in regular CNNs, whereas sparse matrices arise in pruned CNNs.)
   Our processor is intelligent such that when input operands are zero, it does not perform multiplication on them. However, our memory engine is not intelligent, so it fetches even zero elements also.

   (a) Find AI of Case 1 and Case 2.
   (b) Now, assume memory engine is optimized such that only non-zero entries of sparse matrices are fetched from the memory. Also, assume that only non-zero entries need to be written to output (X), since before execution of the above code, the output was initialized to zero. Now, find the AI of Case 2.

10. (a) Binarized AlexNet requires 1.5 billion operations (GOPS) to classify one image. The compute-bound performance of ZU19EG FPGA is 66 TOPS for binary operations. If a design reaches 75% of this peak performance, how many images will it classify in 1 second? (1TOPS = 1000 GOPS) (1 mark)

(b) For both binarized and 8b fixed-point, AlexNet requires 1.5 billion operations (GOPS) to classify one image. 8b fixed-point and binarized AlexNet require 50MB and 7.4MB memory, respectively. Find out arithmetic intensity of AlexNet for both binarized and 8b fixed-point versions. (2 mark)

11. [2 mark] Assume, on KNL machine, peak GFLOP/s is 2199. Bandwidth of L1, L2, MCDRAM and DRAM are 6040, 1827, 372 and 77, respectively (all Bandwidth in GB/s). Find the arithmetic intensity required for achieving peak FLOP on using MCDRAM and on using DRAM.