Odd-even transposition sorting:

Index: 0 1 2 3 4

23 76 89 12 9

I o/e 23 76 89 9 12

e/o 23 76 9 89 12

II o/e 23 9 76 12 89

e/o 9 23 12 76 89

III o/e 9 12 23 76 89

e/o 9 12 23 76 89

F1.cl

\_\_kernel void odd(\_\_global int\* A)

{

int idx=get\_global\_id(0);

int N=get\_global\_size(0);

if((idx%2)!=0 && idx+1<=N-1)

{

if(A[idx]>=A[idx+1])

{

int temp=A[idx];

A[idx]=A[idx+1];

A[idx+1]=temp;

}

}

}

\_\_kernel void even(\_\_global int\* A)

{

int idx=get\_global\_id(0);

int size=get\_global\_size(0);

if((idx%2)==0 && idx+1<=size-1)

{

if(A[idx]>=A[idx+1])

{

int temp=A[idx];

A[idx]=A[idx+1];

A[idx+1]=temp;

}

}

}

Host

Buffer - BufferA

program

cl\_kernel kernel1 = clCreateKernel(program, "odd", &status);

cl\_kernel kernel2 = clCreateKernel(program, "even", &status);

globalWorkSize = N;

for(int i=0;i<N/2;i++)

{

status = clEnqueueNDRangeKernel( cmdQueue, kernel1, 1, NULL, &globalWorkSize,

NULL, 0, NULL, NULL);

status = clEnqueueNDRangeKernel( cmdQueue, kernel2, 1, NULL, &globalWorkSize,

NULL, 0, NULL, NULL);

}

clEnqueueReadBuffer( cmdQueue, bufferA, CL\_TRUE, 0, datasize, A, 0, NULL, NULL);