Work 8 Кириленко Константин 351



void ParallelBitReversing(complex<double>\* inputSignal,

complex<double>\* outputSignal, int size) {

int bitsCount = 0;

//bitsCount = log2(size)

for (int tmp\_size = size; tmp\_size > 1; tmp\_size /= 2, bitsCount++);

//ind - index in input array

//revInd - correspondent to ind index in output array

#pragma omp parallel for

for (int ind = 0; ind < size; ind++)

{

int mask = 1 << (bitsCount - 1);

int revInd = 0;

for (int i = 0; i < bitsCount; i++) //bit-reversing

{

bool val = ind & mask;

revInd |= val << i;

mask = mask >> 1;

}

outputSignal[revInd] = inputSignal[ind];

}

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер теста | Размер входного сигнала | Мин. время работы последовательного приложения (сек) | Мин. время работы параллельного приложения (сек) | Ускорение |
| 1 | 32768 | 0.003 | 0.001 | 3 |
| 2 | 65536 | 0.007 | 0.004 | 1.75 |
| 3 | 131072 | 0.015 | 0.006 | 2.5 |
| 4 | 262144 | 0.033 | 0.011 | 3 |
| 5 | 524288 | 0.07 | 0.025 | 2.8 |

Использование параллельной версии функции BitReversing позволяет еще немного увеличить ускорение.