Work 7 Кириленко Константин 351



Параллельная реализация некотороых функций:

void ParallelFFTCalculation(complex<double>\* signal, int size) {

int m = 0;

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

for (int p = 0; p < m; p++)

{

int butterflyOffset = 1 << (p + 1);

int butterflySize = butterflyOffset >> 1;

double coeff = PI / butterflySize;

#pragma omp parallel for //schedule(dynamic)

for (int i = 0; i < size / butterflyOffset; i++)

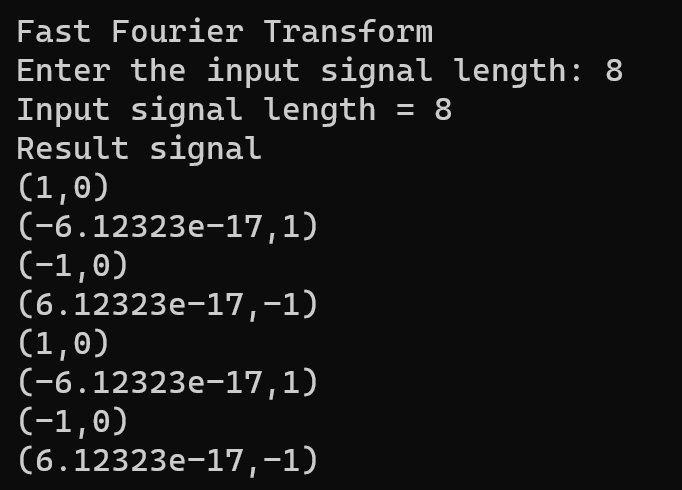
for (int j = 0; j < butterflySize; j++)

Butterfly(signal, complex<double>(cos(-j \* coeff),

sin(-j \* coeff)), j + i \* butterflyOffset, butterflySize);

}

}



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер теста | Размер входного сигнала | Мин. время работы последовательного приложения (сек) | Мин. время работы параллельного приложения (сек) | Ускорение |
| 1 | 32768 | 0.003 | 0.001(0.001) | 3 |
| 2 | 65536 | 0.007 | 0.004(0.003) | 1.75 |
| 3 | 131072 | 0.015 | 0.006(0.006) | 2.5 |
| 4 | 262144 | 0.033 | 0.012(0.02) | 2.75 |
| 5 | 524288 | 0.07 | 0.027(0.046) | 2.593 |

Параллельная версия выигрывает по времени у последовательной примерно в 2.5 раза. Использование метода schedule(dynamic) замедляет выполнение.