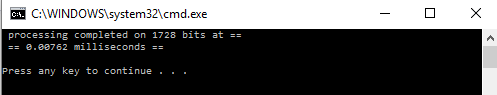
**Timing results of Mapper:**

Mapper using **Intel® Math Kernel Library** (Intel MKL):



Mapper using serial optimized code:



The serial code shows on average equal performance to that using the Intel MKL, but serial code sometimes gives a better performance and reaches. This happens because the processing happens on a low number of bits at large numbers MKL works better and give higher performance than serial code.

**Intel® Math Kernel Library:**

The Intel® Math Kernel Library (Intel® MKL) provides a comprehensive set of math functions that are optimized and threaded to exploit all the features of the latest Intel® processors. Intel MKL improves performance with math routines for software applications that solve large computational problems as Intel MKL linear algebra routines, **fast Fourier transforms**, vectored math functions, random number generation functions, Matrix-Matrix operations and other functionality.

Intel MKL is built using the Intel® C++ and Fortran Compilers and threaded using OpenMP. Its algorithms are constructed to balance data and tasks for efficient use of multiple cores and processors.

**Performance Enhancements:**

The Intel® Math Kernel Library has been optimized by exploiting both processor and system features and capabilities as those routines that most profit from cache-management techniques.

The major optimization techniques used throughout the library include:

* Loop unrolling to minimize loop management costs
* Copying to reduce chances of data eviction from cache
* Data prefetching to help hide memory latency
* Multiple simultaneous operations to eliminate stalls due to arithmetic unit pipelines
* Use of hardware features such as the SIMD arithmetic units, where appropriate

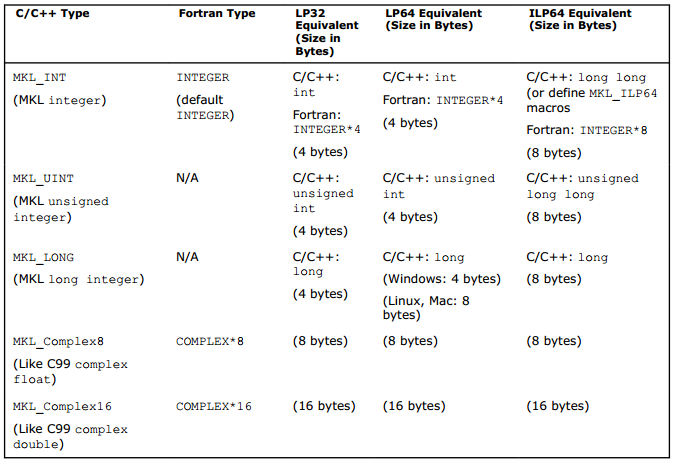
To achieve all the above, the first time a function from the library is called, a runtime check is performed to identify the hardware on which the program is running. Based on this check, a code path is chosen to maximize use of instruction- and-register level SIMD parallelism and to choose the best cache-blocking strategy. Intel MKL is also designed to  
be thread safe, which means that its functions operate correctly when simultaneously called from multiple application threads.

**Parallelism:**

Intel® MKL offers performance gains through parallelism provided by the symmetric multiprocessing performance (SMP) feature. You can obtain improvements from SMP in the following ways:

* One way is based on user-managed threads in the program and further distribution of the operations over the threads based on data decomposition, domain decomposition, control decomposition, or some other parallelizing technique. Each thread can use any of the Intel MKL functions because the library has been designed to be thread-safe.
* Another method is to use the FFT and BLAS level 3 routines. They have been parallelized and require no alterations of your application to gain the performance enhancements of multiprocessing. Performance using multiple processors on the level 3 BLAS shows excellent scaling. Since the threads are called and managed within the library, the application does not need to be recompiled thread-safe.

**C Datatypes Specific to Intel MKL:**



**Code Samples:**

The following is a sample of the mapper using Intel® MKL: