Parallel computing Ex1:

Ofir Birka

Bar Vered

1. Maximum speed - 2.10 Ghz

   L1 Cache – 128 Kb

   L2 cache – 512 Kb

   L3 cache – 3 M

   RAM – 5.9 Gb

   The manufacture claims that the CPU can do 2.5 instructions in one cycle. The speed is 2.1 GHz, therefore the cpu can do 5.25*10^9 instructions in one second (IPS).

2. We are checking this claims with a while loop that decreases to see how much time it takes the computer to finish the loop.

   We wrote this code to verify the performance:

```
register long i = 6000000000L;
while( i-- )
    ;
```

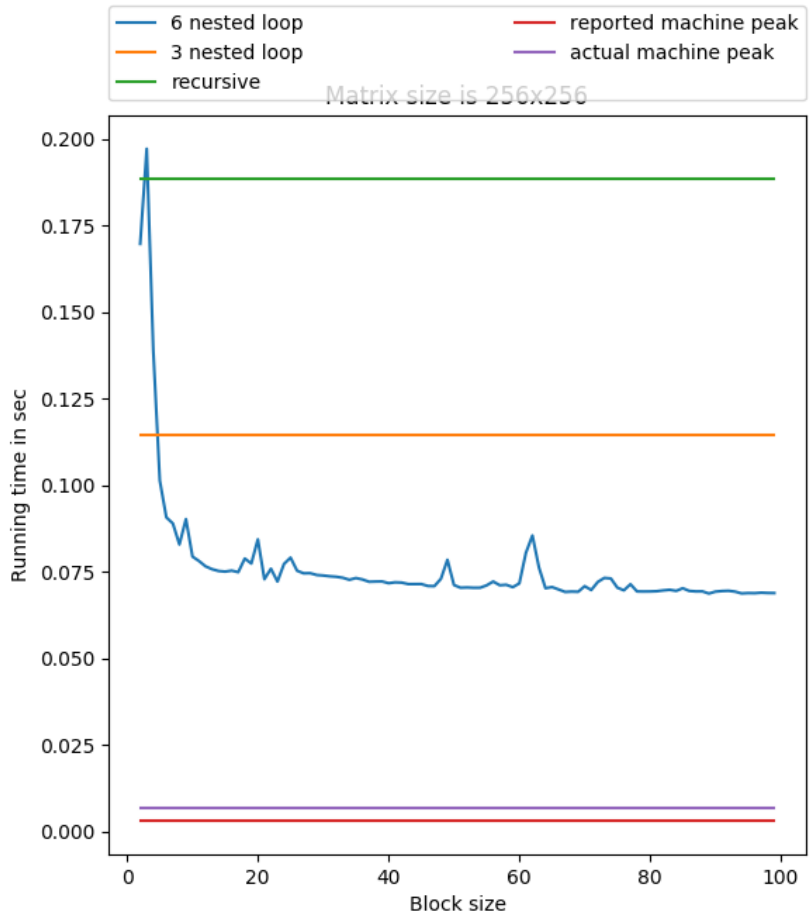
   We concluded that it does 2.4*10^9 IPS.

3. Coding

4. Reported Machine Peak – it does 5.25*10^9 IPS and we need to do 256^3 multiplications, which means the program should run in *0.00319566 seconds*

   Actual machine peak - *0.006990507 seconds*

   The graph:



5. We can see that the recursive algorithm is worst, then we can see the iterative (3 nested loops) algorithm is better than the recursive one, and the 6 nested loops is dependent on the block size, a smaller block size makes it almost like the 3 nested loops algorithm and this makes sense, once we give a larger block size, the time it takes to run the 6 nested loops decreases and becomes the best algorithm for matrix multiplication.

   Conclusions: The recursive algorithm uses a lot of stack unrolling and therefore is the worst algorithm. The naive algorithm is in the middle algorithm in performance, its runtime is stable. The 6 nested loops is the best algorithm, we can peaks and decreases on certain block sizes, and the bigger the block size is, the better performance of the multiplication is, until a certain value is reached and can no longer can beneath it. We used a matrix 256x256 in order to show the efficiency of the 6 nested loop because it is an algorithm that is “easy” on the cache, and needs enough elements in order to test that.