Journal of Circuits, Systems, and Computers
Vol. 26, No. 5 (2017) 1750086 (13 pages)

DOI: 10.1142/S0218126617500864

World Scientific
www.worldscientific.com

# An Analytic Method for Estimating the Computation Capacity of Computing Devices*

Boris Ryabko†
*Laboratory of Data Protection, Institute of Computational Technologies of SB RAS, Novosibirsk State University, Novosibirsk 630090, Russia*
*boris@ryabko.net*

Anton Rakitskiy
*Department of App. Math and Cybernetics, Siberian State University of Telecommunications and Informatics, Kirova St., 86, Novosibirsk 630102, Russia*
*rakitsky.anton@gmail.com*

Received 16 July 2015
Accepted 26 October 2016
Published 22 December 2016

A recently suggested method for evaluating computer performance is applied to real processors. The method is based on notion of Computer Capacity, which is determined only by the computer architecture and can be computed theoretically on the design stage. The computer capacity for different Intel and AMD processors is calculated and the results are compared with those of widely recognized benchmarks. The obtained results show that computer capacity is a reasonable estimate of computer performance.

*Keywords*: Computer capacity; performance of computers; information theory.

## 1. Introduction

Currently used methods of performance evaluation cannot be considered completely objective, since they have some important limitations. The widely used methods of evaluating computer performance are based on benchmarks. Here, a benchmark is a test set of tasks, which helps to determine the resources, such as the execution time, the amount of memory, etc, required to solve them. The comparison of computers is performed based on the analysis of these resource requirements.

*This paper was recommended by Regional Editor Piero Malcovati.
†Corresponding author.

A major limitation is that one cannot use benchmarks in order to evaluate a computer during the development phase, when there is no working model. Here, by the development phase we implied the process of the design of a new processor and the manufacturer needs to evaluate the performance of developed processor during this process. This imposes restrictions on the usage and increases the cost of evaluation and comparison of computers. Moreover, the objectiveness of benchmarks is reduced by the fact that they are focused on specific tasks. For example, a computer can be evaluated based on the speed of floating point operations, or based on the speed of processing multimedia information. All tests are specific and cannot give an overall evaluation. Let us emphasize that the problem of the computers performance evaluation attracts the attention of many researchers.[1,2]

In the preceding work,[3] a new theoretical approach to performance estimation of computers and computing systems was proposed. This method is based on the notion of computer capacity, which relies on the concept of Shannon entropy, lossless channel capacity and some other ideas of Information Theory. The computer capacity is fully defined by the processor architecture (the set of instructions, memory organization and size, CPU frequency, number of cores).[3] A working model is not needed, so that the evaluation can be performed at the design stage.

However, in the preceding[3] estimations of the computer capacity for real computers were not provided. In this paper, the computer capacity is estimated for the most common Intel processors and some AMD processors. Then, the data are compared with those obtained on some established benchmarks. The results show that the computer capacity provides reasonable estimates of the computer performance.

The outline of the paper is as follows. Section 2 describes the theoretical basis of the suggested method. In Sec. 3, we describe some of the real processors features which are important to the estimation of the computer capacity. The obtained results with explanations and some analysis are presented in Sec. 4 and, finally, in Sec. 5 we present the summary analysis of all the obtained results and benchmarks and make the conclusions.

## 2. Computer Capacity

### 2.1. *Main concept and definitions*

Here we present only a summary of the main concept. A model of a computer[3] includes a set of processor instructions $I$ and the available memory $M$. It is important to note that each instruction $x \in I$ contains not only the name of the instruction itself, but also the operands (memory addresses, indexes of registers, etc.). Thus, if we examine two instructions MOV, for example, that work with different memory locations, $I$ would contain both of them as independent and different instructions. Processor task $P$ is defined as a sequence of instructions $X(P) = x_1 x_2 x_3 \dots x_n, x_i \in I$. Accordingly, if a task contains a loop which repeats five times,

then the sequence $X$ would contain the body of that loop repeated five times. We must take into account that not all sequences of instructions are allowed. Generally speaking, there are some pairs of instructions which cannot be executed by the processor. In other words, it is possible that sequences of instructions that can be executed have to obey some rules. We define $S_c$ as the set of all admissible sequences, and consider any two different sequences of processor instructions from set $S_c$ as two different processor tasks. Consequently, any task can be represented as a sequence of instructions from $S_c$.

Let us denote the execution time of instruction $x$ by $\tau(x)$. In order to simplify, we suppose that all execution times $\tau(x), x \in I$, are integers and the greatest common divisor of $\tau(x), x \in I$, equals 1 (This assumption is valid for most of processors if the time unit equals the so-called clock rate and there are instructions whose execution time is one unit, i.e., $\tau(x) = 1$). Then the execution time $\tau(X)$ of a sequence of instructions $X = x_1 x_2 x_3 \dots x_t$ is given by

$$\tau(X) = \sum_{i=1}^{t} \tau(x_i)\,. \tag{1}$$

Let us denote the number of different tasks whose execution time equals $T$ by $N(T)$. Then, the computer capacity is defined as follows:

$$C(I) = \lim_{T \to \infty} \log_2 N(T)/T \tag{2}$$

bits per time unit.

Now we can try to explain the main idea of the suggested approach. Let us imagine that there is a processor which can execute $N$ different sequences of instructions in, say, one hour. Then, roughly speaking, this processor can execute $N^2$ sequences of instructions in two hours, because if $s_1$ and $s_2$ are any one-hour sequences, then the combined sequence $s_1 s_2$ is a two-hour one. Analogously, approximately $N^k$ sequences can be executed in $k$ hours. So, the number of possible sequences (and solvable tasks) grows exponentially as a function of the time $t$, thus $\log N(t)/t$ (or the limit of this value) appears to be an adequate measure of the processor capacity. In other words, the number of possible sequences grows exponentially with the rate that is asymptotically equal to the capacity of the processor.

### 2.2. *Methods of estimating the computer capacity*

The easiest way to estimate the computer capacity is to suppose that all sequences of instructions are allowed. In other words, we consider set of instructions $I$ as an alphabet and assume, that all sequences from that alphabet can be executed. Thus, let there be a processor with set of instructions $I$, whose execution time is $\tau(x), x \in I$, and all sequences of instructions are allowed. (In other words, if we assume that $I$ is an alphabet, then all words from symbols of this alphabet can be considered as allowed sequence of processor instructions.) Our goal is to find a number of sequences whose length is $t$, i.e., $N(t)$, $t > 0$. This problem is well known in combinatorial

analysis (see, for example, work[4]). In particular, it is known that $N(t)$ grows exponentially if $t$ goes to infinity. In other words, $N(t) = 2^{Ct}$. We estimate the number of all sequences of instructions. Obviously, it is the upper bound of the computer capacity. On the other hand, in fact, it is the exact estimation for most of modern processors, because in modern computers any instructions can be executed after any other.

The question we consider now is how one can calculate (or estimate) the capacity for this case. The method of calculation is known in combinatorial analysis[4] and was used by Shannon.[5] In the considered case the following equation

$$N(t) = N(t - \tau(x_1)) + N(t - \tau(x_2)) + \cdots + N(t - \tau(x_S)) \tag{3}$$

is valid and, according to a result in finite differences, the capacity $C(I)$ is equal to the logarithm of the largest real solution $X_0$ of the following equation:

$$X^{-\tau(x_1)} + X^{-\tau(x_2)} + \cdots + X^{-\tau(x_S)} = 1, \tag{4}$$

where $I = x_1, \ldots, x_S$ is the set of processor instructions and $\tau(x_i), x_i \in I$, is the execution time of the instruction $x_i$. In other words, $C(I) = \log_2 X_0$, see Ref. 5. Note that the solution can be found by the so-called bisection method, which is the simplest root-finding algorithm. We used such a program of calculating the computer capacity that finds the solution with precision equal to $10^{-10}$.

## 3. Application of Method on Real Processors

In this section, we describe some features that are used in the construction of the characteristic equations for real processors. We chose Intel and AMD processors because the information about them is open and freely available. We calculate computer capacity of the following processors: Intel 80486, Pentium, Pentium MMX, Pentium Pro, Pentium II, Pentium III, Pentium IV, Pentium M, Intel Core Solo/Duo, Intel Core 2 Solo/Duo, AMD K10. In the following subsections, we describe features of equations construction for each processors from that list.

### 3.1. *Intel 80486*

The Intel 486 was introduced in 1989 and was a higher performance follow-up to the Intel 80386. That was the first processor composed of a pipeline and two levels of cache memory. The first-level cache is posed at processor's crystal. Its size was 8 KB in the first models and 16 KB in the later models. Its frequency is equal to the processor frequency. This processor also has a second-level cache, which is larger in size than first-level cache, but significantly slower. The presence of cache memory greatly accelerates the access to frequently used memory cells. A detailed description of calculating the computer capacity for processors with cache memory can be found in preceding paper.[3] By solving the resulting characteristic equation, we obtain $X_0 \approx 1734.13$, $C(I) \approx \log_2 1734.13 \approx 10.76$ bit/cycle.

### 3.2. *Intel P5 processors (Pentium, Pentium MMX)*

The Intel Pentium processor was first introduced on March 22, 1993. The P5 microarchitecture was Intel's fifth generation and the first superscalar IA-32 one. This processor has an additional second execution pipeline, which allows it to achieve super-scalar performance (two pipelines, known as u and v, can execute two instructions per clock together). The next processors in the Pentium family introduces Intel MMX technology (the Pentium Processor with MMX technology). Intel MMX technology uses the single-instruction multiple-data (SIMD) execution model to perform parallel computations on packed integer data contained in 64-bit registers.

This processor is super-scalar and can (in the ideal case) run two integer instructions at the same time using the additional pipeline. However, this feature has some limitations. The first one is that v-pipeline can run only certain types of instructions. The second limitation is that not all instructions can be paired with one another. To be more precise, all instructions are divided into four groups:

(i) U-pairable. These instructions run only at u-pipeline, but can be paired with another instruction which runs on the v-pipeline.
(ii) V-pairable. These instructions run at both pipelines, but can be paired only if run at the v-pipeline.
(iii) UV-pairable. These instructions run at both pipelines and can be paired at both pipelines.
(iv) NP(not pairable). These instructions run only at the u-pipeline and cannot run in pair with other instructions.

Detailed information about the method of calculating the computer capacity for these processors can be found in preceding papers.[6,7] After constructing characteristic equation and solving it, we obtain $C(I) \approx 25.56$ bit/cycle. For the MMX processor the value of computer capacity is $C(I) \approx 28.35$ bit/cycle.

### 3.3. *Intel P6 processors (Pentium Pro, Pentium II, Pentium III)*

The sixth-generation Intel x86 micro-architecture, implemented by Pentium Pro microprocessor, was first introduced in November 1995. The succeeding models based on this microarchitecture were presented in 1997 (Pentium II) and 1999 (Pentium III). P6 processors architecture radically differs from P5. Most of modern Intel processors are based on this architecture, so it requires a detailed review. That processor pipeline is composed of several base blocks:

(i) branch prediction
(ii) instruction fetch unit
(iii) instruction decoder
(iv) register alias table, register renaming

(v) $\mu$op re-ordering buffer
(vi) reservation station
(vii) ports connecting to execution units
(viii) write-back of results to re-ordering buffer
(ix) register retirement file

More detailed information about each block is available in detailed description of microarchitectures,[8] but here we just consider the main components needed for our method. The fetch instruction unit can read 16 bytes in 1 cycle, so it generates more than 1 instruction per cycle. The decoding unit consists of three decoders, each of which converts instructions and divides them into $\mu$ops (micro-operations). The first decoder can process instructions which are divided into four or less $\mu$ops for one cycle. The second and the third decoders can process instructions which generate one $\mu$op. So, in the ideal case, we get 6 $\mu$ops for one cycle. Herewith, in the worst case, it is 2 $\mu$ops. Capacity of all next units is 3 $\mu$ops per cycle. We calculate an upper bound of the computer capacity, so we assume that the worst case is impossible and our pipeline capacity is 3 $\mu$ops per cycle.

Since each $\mu$op is performed for one clock cycle, we assume that instruction execution time is the number of $\mu$ops which it generates. So, if three $\mu$ops can be executed concurrently, we assume that pipeline runs three independent threads. It means that we need to multiply total value of $C(I)$ by 3, to calculate correct computer capacity. Detained information about $\mu$ops generated by each instruction can be found in the instructions list.[9] It is important to note how the computer capacity estimates for Pentium II and Pentium III. For Pentium II we add all MMX instructions into the instruction set. For Pentium III we also add the SSE instructions. After solving the equation obtained this way, we get $C(I) \approx 36.62$ bit/cycle for Pentium Pro, $C(I) \approx 37.69$ for Pentium II and $C(I) \approx 42.02$ for Pentium III.

### 3.4. *Intel NetBurst processors (Pentium IV)*

The NetBurst micro-architecture was the successor to the P6 microarchitecture. The first microprocessor to use this architecture was Pentium IV, released on November 2000 and all subsequent Pentium IV and Pentium D variants have also been based on this microarchitecture. The Intel Pentium IV processors are very different from the design of other Intel processors. Now, it is no longer used in new designs, because of its lower efficiency. The primary goal of the NetBurst microarchitecture is to obtain the highest possible clock frequency. This is achieved by adding to the pipeline an additional stage. In contrast to P6 microarchitecture, which has 9 stages in pipeline, NetBurst has 20 stages.

However, computing the estimation of computer capacity for this processor is not much different from computing it for the P6 processors, as the capacity of NetBurst pipeline is still three micro-operations per cycle. So, solving the equation gives us $C(I) \approx 39.66$ bit/cycle.

### 3.5. *Intel PM processors (Pentium M, Core Solo, Core Duo)*

The PM is a family of mobile x86 processors first introduced by Pentium M in March 2003 (Core Solo and Core Duo were launched in January 2006). Basically, PM has the same architecture as P6. The main stages in the pipeline are: branch prediction, instruction fetch, register renaming, etc. Several minor modifications are made, but the overall functioning is almost identical to P6. However, there are some modifications that affect the calculations using our method.

First modification is the stack engine. Stack instructions such as PUSH, POP, CALL and RET all modify the stack pointer ESP. All $\mu$ops that modify only stack pointer are executed in stack engine instead of the execution unit. The second modification is the micro-operations fusion. The register renaming and retirement stages in the pipeline are bottlenecks with the maximum throughput of 3 $\mu$ops per clock cycle. The fusion technique allows this processor to accelerate the throughput by joining some micro-operations together. The $\mu$op fusion technique can be applied only to two types of combinations: memory write $\mu$ops and read-modify $\mu$ops. We can assume that fused $\mu$ops are executed simultaneously so we consider it as a single $\mu$op at all stages. When we construct the characteristic equation with all these features and solve it, we obtain $C(I) \approx 51.198$ bits/cycle.

### 3.6. *Intel Core 2 processors (Core 2 Solo, Core 2 Duo)*

The Core 2 Solo, introduced in September 2007, is the successor to the Core Solo. The microarchitecture Intel Core 2 is based on PM architecture. The pipeline is expanded to handle 4 micro-operations per clock cycle. The second important feature of these processors series is the introduction of processors with several cores. The method of calculating computer capacity for processors with several cores is described in detail in the preceding paper.[3] So, the most important difference for our method is the throughput in the pipeline. In PM processors this value is 3 $\mu$ops per clock cycle and here it increases to 4 $\mu$ops. It means that we should multiply the value of $C(I)$ by 4 instead of 3 at the final calculation stage. So, after finding the solution of the equation we get $C(I) \approx 68.847$ bits/cycle for Core 2 Solo and $C(I) \approx 137.69$ for Core 2 Duo.

### 3.7. *AMD K10 processors (Phenom, Opteron)*

The AMD K10 is a microarchitecture by AMD based on the K8 micro-architecture. The first microprocessors from this family, quad-core third-generation Opterons, were introduced on 10 September 2007. AMD processors have fundamentally different architecture. Intel processors split instruction into micro-operations and their parallelism is based on the fact that different types of $\mu$ops are performed by different execution units. In contrast, AMD processors are composed of several independent and parallel pipelines. It provides a univocal parallel instruction execution, but

considerably increases the execution time of each instruction. In our case, AMD K10 processor consists of three pipelines. Actually, despite the increased instruction execution time, the architecture of this processor is very simple, which allows for a considerable increase in its clock frequency.

So after constructing the equation for this architecture, we can solve it and find $C(I) \approx 211.1924$ bits/clock for AMD Phenom 9500 X4 processor.

### 3.8. *Analysis of the actual characteristic equation*

We use Pentium II for a detailed analysis of its equation in the original form contains only 404 terms. The characteristic equation "Eq. (4)" for Pentium II processor after reducing the similar terms is as follows:

$$\begin{aligned}
&\frac{6647}{X^1}+\frac{67811}{X^2}+\frac{672150}{X^3}+\frac{2711834}{X^4}+\frac{712723}{X^5}+\frac{11591715}{X^6}\\
&\quad+\frac{42467328}{X^7}+\frac{11960448}{X^8}+\frac{13631495}{X^9}+\frac{2293761}{X^{10}}+\frac{2760393249}{X^{11}}\\
&\quad+\frac{10840612864}{X^{12}}+\frac{2922422272}{X^{13}}+\frac{3456385025}{X^{14}}+\frac{543817728}{X^{15}}\\
&\quad+\frac{2282356738}{X^{16}}+\frac{537264130}{X^{17}}+\frac{805306371}{X^{18}}+\frac{67108865}{X^{19}}\\
&\quad+\frac{1780482048}{X^{20}}+\frac{167772161}{X^{21}}+\frac{4362076176}{X^{22}}+\frac{536870912}{X^{25}}\\
&\quad+\frac{2}{X^{26}}+\frac{1}{X^{31}}+\frac{4294967296}{X^{33}}+\frac{1}{X^{34}}=1
\end{aligned}$$

Let us take a closer look at some terms of this equation. The first term $6647/X^1$ consists of the following instructions (in brackets we indicated the number of different instructions): mov r r/i (272), mov r sr (96), movsx r r (256), movzx r r (256), lahf (1), sahf (1), add r r/i (272), sub r r/i (272), and r r/i (272), or r r/i (272), xor r r/i (272), cmp r r/i (272), test r r/i (272), inc r (16), dec r (16), neg r (16), not r (16), aaa (1), aas (1), daa (1), das (1), cbw (1), cwde (1), cwd (1), cdw (1), sht r (16), shl r (16), sar r (16), ror r (16), rol r (16), bt r r/i (272), btr r r/i (272), bts r r/i (272), btc r r/i (272), SETcc r (16), jmp short (256), jmp r (16), conditional_jump short (256), clc (1), stc (1), cmc (1), long_nop_0F_1F (1), movd r r (256), movq r r (256), padd mm mm (64), psub mm mm (64), pcmp mm mm (64), pmull mm mm (64), pmadd mm mm (64), pand_N mm mm (64), por mm mm (64), pxor mm mm (64), psra mm mm/i (72), psrl mm mm/i (72), psll mm mm/i (72), pack mm mm (64), punpack mm mm (64), pmovmskb r mm (128), pshufw mm mm i (64), pinsrw mm r i (128), pavgb mm mm (64), pavgw mm mm (64), pmin mm mm (64), psw mm mm (64), pmaxub mm mm (64), pmulhuw mm mm (64), where the operands are r-register, i-immediate data, sr-segment register, mm-64 bit mmx register.

The second term $67811/X^2$ consists of the following instructions: CMOVcc r r (256), xlat (1), adc r r/i (272), sbb r r/i (272), rcr r (16), rcl r (16), shld r r i (256), shrd r r i (256), bsf r r (256), bsr r r (256), jmp near (65536), j_e_cxz short (256), lods (1), bswap r (16), sfence (1) and pextrw r mm i (128).

The rest of the terms are computed similarly.

## 4. Analysis of the Results

A direct comparison of all the processors described presents several problems. The first examined processor Intel 80486 was constructed in 1989 and the last processor AMD Phenom X4 in 2008. It is a long time period, so we do not have a common benchmark to compare all of them. We need to divide all processors into three overlapping groups. All the characteristic equations of these processors and programs which are used for building and calculating the computer capacity can be found at Ref. 10. First group contains I80486, Pentium and Pentium MMX processors are compared against the ICOMP benchmark. ICOMP index is the benchmark that was used by Intel to estimate the performance of its processors and it combines the values of several benchmarks.[11] As the measurement units for benchmarks and the computer capacity differ, we divide the values corresponding to examined processor by the values of previous one. In this way we get the relative value without the measurement units at $y$-axis. As we can see in Fig. 1, both characteristics have similarities, although there are some deviations. This is because the benchmark is composed of a finite set of specific tasks that developers have considered as most important. In contrast, the computer capacity evaluates the ability of the considered computers to solve all possible tasks.

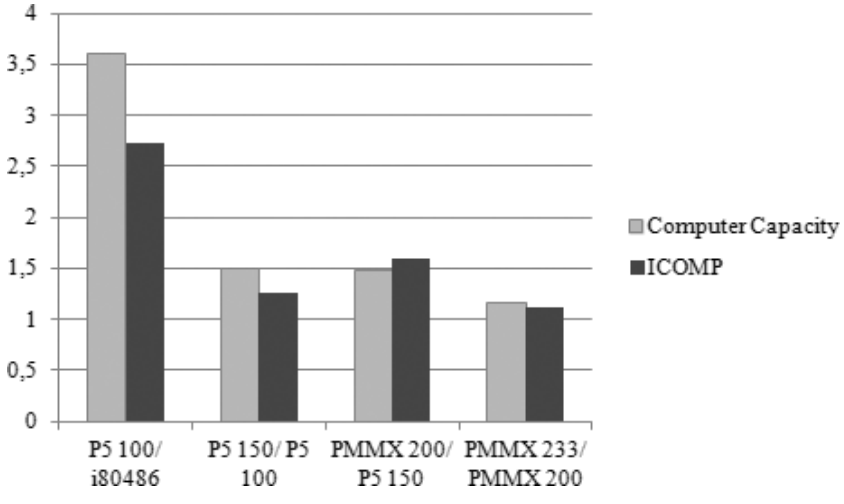


Fig. 1. Comparison of processors from the first group.

Table 1. Group 2: Values of computer capacity & benchmarks.

| CPU | Clock rate, MHz | Computer capacity, Mbits/s | ICOMP | SPECint95 | SPECfp95 |
|---|---|---|---|---|---|
| Pentium | 150 | 3834 | 114 | 4.13 | 3.58 |
| PMMX | 200 | 5670 | 182 | 6.37 | 4.87 |
| PPRO | 200 | 7323.68 | 220 | 8.59 | 6.34 |
| PII | 400 | 15,076.55 | 440 | 15.6 | 12.9 |
| PIII | 500 | 21,011.17 | 644 | 21.6 | 16.2 |

The next group contains processors from Intel Pentium to Intel Pentium III (P5 and P6 microarchitectures). In the second group we examine processors with P5 and P6 microarchitectures. It is especially interesting because these microarchitectures are fundamentally different. Here we use three benchmarks: ICOMP, SPECint95 and SPECfp95. The ICOMP benchmark was described above, so it is necessary to explain what is SPEC. The Standard Performance Evaluation Corporation (SPEC) is a nonprofit organization whose main task was to create a standardized set of benchmarks that can be applied to the newest generation of computers.[12] This benchmarks are widely used for the performance evaluation even today.[13] More detailed descriptions of these benchmarks can be found in their specifications.[14,15] Here, in Table 1 the values of all described benchmarks and the computer capacity are presented. As the measurement units for these values are different we build the graph at Fig. 2 in the following way. Let us take the values of the first processor (Pentium 150) in Table 1 as measurement unit, so we just divided

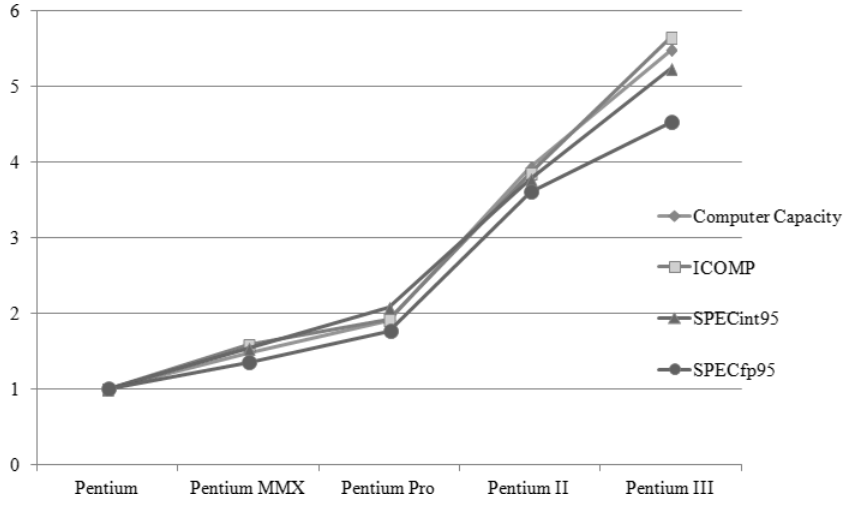


Fig. 2. Comparison of processors from the second group.

Table 2. Group 3: Values of computer capacity & PassMark.

| CPU | Clock rate, MHz | PassMark | $C(I)$, bits/clock | Computer capacity |
|---|---|---|---|---|
| Pentium 3 | 1266 | 270 | 42.02 | 53,200.28 |
| Pentium M | 2000 | 495 | 51.2 | 102,396.06 |
| Core Solo T1350 | 1860 | 443 | 51.2 | 95,228.34 |
| Pentium 4 | 2600 | 405 | 39.66 | 103,115.51 |
| Core 2 Solo U3500 | 1400 | 479 | 68.85 | 96,385.62 |
| Core 2 Duo T7300 | 2000 | 1232 | 137.69 | 275,387.48 |

the each value in table by the corresponding values of Pentium 150 and build the graph with the obtained results. In this case we avoid any measurement units at the $y$-axis. As we can see at Fig. 2, the obtained data are well correlated with the results of the considered benchmarks.

The last group contains all processors from Pentium III to AMD Phenom X4. The technical characteristics and the results of Computer capacity and Passmark benchmark for Intel processors are given in Table 2. In the third group we are using PassMark benchmark[16] for comparison because the processors were produced over a long time interval and there are few suitable benchmarks for comparing all the processors presented. The plot in Fig. 3 is built relative to the Pentium 3 values that we take as 1 (the same way as for the graph at Fig. 2). In Fig. 3 we can see that the

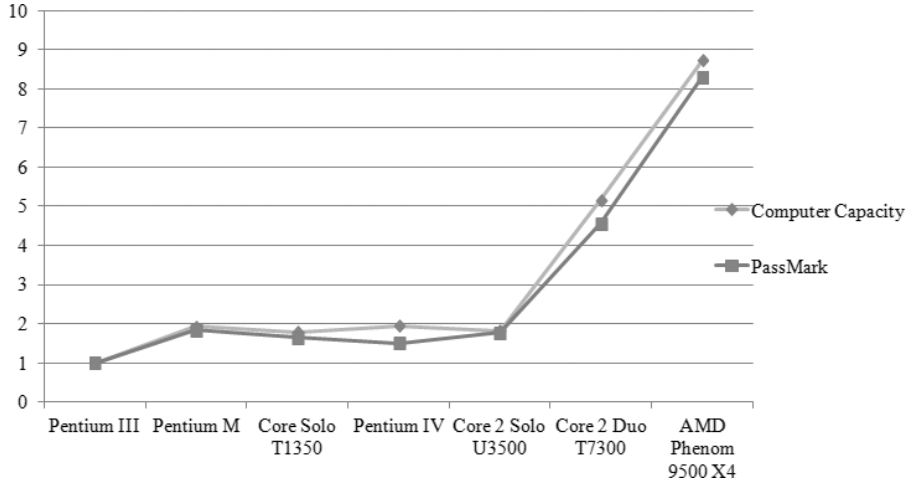


Fig. 3. Comparison of processors from the third group.

computer capacity characteristic has some deviations from the benchmark, but they are insignificant and can be explained by the subjectiveness of the benchmark.

## 5. Conclusion

To confirm the effectiveness of the suggested method we have analyzed the published data which are actually based on the values of benchmarks. Such a comparison is more objective than if we carried out the experiments by ourself. In order to improve the objectiveness we considered the most popular benchmarks developed by different authors and compared the suggested estimation method to each of them. The results which were presented in Fig. 2 and Fig. 3 show that the computer capacity is a reasonable characteristic and is consistent with the existing data.

In our paper, we used the published data of the ICOMP,[11] SPEC[14,15] and PassMark[16] benchmarks. All considered benchmarks use some sets of programs (each benchmark uses its own set) and form the results on the basis of the execution of its set of programs on the investigated processor.[1,2,17] In contrast to benchmarks, the suggested method does not require a working model (physically constructed processor or its simulation) of the investigated processor and the computer capacity can be estimated without any experiments over such working models.

All the presented results show us that the computer capacity is an accurate characteristic and could be used for comparison of the performance of processors. Also we need to emphasize that our characteristic can be used by the developers of a new processors at the design stage to estimate the performance of a developed processor without the construction of its working model.

## Acknowledgments

The authors were supported by the Russian Foundation for Basic Research, Grant no. 15-07-01851.

## References

1. D. J. Lilja, *Measuring Computer Performance: A Practitioner's Guide* (Cambridge University Press, 2005).
2. R. P. Weicker, A detailed look at some popular benchmarks, *Parallel Compu.* **17**(10) (1991) 1153–1172.
3. B. Ryabko, An information-theoretic approach to estimate the capacity of processing units, *Performance Eval.* **69** (2012) 267–273.
4. J. Riordan, *Introduction to Combinatorial Analysis* (Courier Corporation, 2012).
5. C. E. Shannon, A mathematical theory of communication, *Bell Sys. Tech. J.* **27** (1948) 623–656.
6. A. Fionov and B. Ryabko, Estimating the performance of computer systems through computer capacity, *Problems of Redundancy in Information and Control Systems (RED), 2012 XIII International Sympo.* (2012), pp. 74–77.

7. A. Fionov, A. Rakitsky and B. Ryabko, Evaluation of computer capacity for p5 intel processors, in *Problems of Redundancy in Information and Control Systems (RED), 2012 XIII International Symp.* (2012), pp. 70–73.
8. A. Fog, The microarchitecture of Intel, AMD and VIA CPUs, Copenhagen University College of Engeneering, www.agner.org/optimize/October 2013.
9. A. Fog, Lists of instruction latencies, throughputs and microoperation break-downs for Intel, AMD and VIA CPUs, Copenhagen University College of Engeneering, Available at www.agner.org/optimize/October 2013.
10. Characteristic equation and all supporting programs. Web: https://goo.gl/MHneDr
11. Intel Corporation, iCOMP Index 2.0 Performance Brief, Available at http://download.intel.com/design/PentiumII/perfbref/24339301.PDF.
12. J. Reilly, A brief introduction to the spec cpu95 benchmark. IEEE-CS TCCA Newsletter. June 1996.
13. A. Phansalkar, Measuring program similarity: Experiments with SPEC CPU benchmark suites, *IEEE Int. Symp. Performance Analysis of Systems and Software, ISPASS 2005*, pp. 10–20.
14. Standard Performance Evaluation Corporation, SPEC CINT95 Benchmarks, Available at https://www.spec.org/cpu95/CINT95/.
15. Standard Performance Evaluation Corporation, SPEC CFP95 Benchmarks, Available at https://www.spec.org/cpu95/CFP95/.
16. PassMark performance test description, Available at http://www.passmark.com/support/performancetest/interpreting_test_results.htm.
17. D. Bhandarkar and D. Jason, Performance characterization of the Pentium Pro processor, *IEEE Third Int. Symp. High-Performance Computer Architecture* (1997), pp. 288–296.