Вычислительные технологии, 2020, том 25, № 6, с. 104–115. © ФИЦ ИВТ, 2020 ISSN 1560-7534
Computational Technologies, 2020, vol. 25, no. 6, pp. 104–115. © FRC ICT, 2020 eISSN 2313-691X

DOI:10.25743/ICT.2020.25.6.007

Information theory as a means of determining the main factors affecting the processors architecture

Rakitskiy Anton. A.[1,2,3,*], Ryabko Boris Ya.[1,2]
[1]Novosibirsk State University, Novosibirsk, 630090, Russia
[2]Federal Research Center for Information and Computational Technologies, 630090, Novosibirsk, Russia
[3]Siberian State University of Telecommunication and Information Sciences, 630009, Novosibirsk, Russia
[*]Correspoding author: Rakitskiy Anton A., e-mail: rakitsky.anton@gmail.com

Received June 15, 2020, revised October 22, 2020, accepted October 30, 2020.

In this article we are investigating the computers development process in the past decades in order to identify the factors that influence it the most. We describe such factors and use them to predict the direction of further development. To solve these problems, we use the concept of the Computer Capacity, which allows us to estimate the performance of computers theoretically, relying only on the description of its architecture.

Keywords: Computer Capacity, processors architecture, information theory, performance processors characteristics, Shannon theory.

Citation: Rakitskiy A.A., Ryabko B.Ya. Information theory as a means of determining the main factors affecting the processors architecture. Computational Technologies. 2020; 25(6):104–115. DOI:10.25743/ICT.2020.25.6.007.

Introduction

Currently, there is a wide variety of processors and computing devices with them in information technologies. The main purpose of this work is to analyze the process of their development and to identify the factors that can be used to predict and control the "evolution" of processors. To achieve this, we propose to use the Computer Capacity characteristic. The main concept of the Computer Capacity characteristic was originally proposed in 2012 [1]. There was shown that using this characteristic we can estimate the performance of any computer theoretically, based solely on the description of its architecture. In other words, the evaluation of the Computer Capacity does not require a working model of the investigated computer. In the following papers [2, 3] it was shown that the presented characteristic is consistent with the results of generally accepted benchmarks [4–7] for processors of different types. In our works we investigated the majority of processors types, such as Intel, AMD, ARM, MIPS and graphical processors as well as supercomputers built on them. The results of presented research have shown that the Computer Capacity characteristic is fairly accurate estimation for the computers performance and, due to the possibility of theoretical evaluation, it can be used at the computer development design stage. In addition, the characteristic was successfully applied to analyze the evolution of Intel processors [8]. In the

course of this study, an evolutionary series of Intel processors over the past 20 years was considered and parameters which have a non-linear effect on processor performance were identified. It was shown that in practice, these parameters were increased by manufacturers when moving from the old processor to the new one. It should be noted that the Computer Capacity allows not only to reveal such parameters, but also to quantify their influence.

In this paper, we show how the Computer Capacity allows us to identify the main factors that determine the trends in the development of processors architectures. The main parameters that determine the development of architectures in the past and in the future are highlighted. Moreover, the Computer Capacity allows not only to highlight these factors, but also to quantify their impact on various architectures. All such factors can be divided into:

- The factors whose influence is obvious, such as number of cores, clock frequency, etc.
- The factors whose influence is not so obvious are the number of threads (by threads we mean the processor's ability to execute independent instructions in parallel), the sizes and the access times of different type of memory, and the set of instructions.

In this article, we focus on the factors of the second group and show how the future development will depend on them.

The preliminary version of this paper was presented at 2019 XVI International Symposium "Problems of Redundancy in Information and Control Systems" (REDUNDANCY) [9].

1. The Computer Capacity

In this section we briefly describe the main definitions and a summary of the theory to understand the concept of the Computer Capacity. The entire theoretical basis with the description and proof of the theorems can be found in [1].

In general, a computer can be represented as the set of instructions I and the memory M to which these instructions address. An instruction $z \in I$ is a combination of the instruction name and the values of its operands. It means that, for example, instructions *mov ax bx* and *mov dx ax* are different and both are included in the set I. We consider computer task Z as the sequence of instructions $Z = \{z_1, z_2, \ldots, z_n\} \in I$. Let us note that if there is a loop in a task which is repeated k times the instructions from the body of this loop are included k times in Z. We denote the execution time of instruction $z \in I$ as $\tau(z)$ and the execution time of computer task $Z = \{z_1, z_2, \ldots, z_n\}$ as the sum of instructions execution times $\tau(Z) = \sum_{i=1}^{n} \tau(z_i)$.

Let us consider the number of all possible computer tasks which execution time equals to T as $N(T) = |Z : \tau(Z) = T|$. In [1] there was shown that this number grows exponentially as the function of time. If there are N_1 tasks with execution time 1 minute, there are N_1^2 tasks with execution time 2 minutes, and correspondingly there are $N_k \approx N_1^k$ tasks which are executed in k minutes. In this way $N(T) \approx 2^{CT}$, where C is the measure which we call the Computer Capacity. So the Computer Capacity can be considered as follows:

$$C(I) = \lim_{T \to \infty} \frac{\log N(T)}{T}. \tag{1}$$

The main task here is how to estimate the value of $C(I)$ from (1). The direct calculation of this limit is impossible, but there exist the method of calculation $C(I)$ in combinatorial analysis. Here we consider the set of instructions I as an alphabet and assume that all

computer tasks Z are words over this alphabet. We will consider that all computer tasks can be executed, i. e. all words over the alphabet I are possible. This approach allows us to estimate the upper bound of the Computer Capacity, because for any computer the set of its permissible tasks is the subset of all possible tasks. In addition we assume that all execution times are integers and their greatest common divisor is 1. We should clarify that this assumption is valid for the majority of processors: instructions execution times are mainly measured in processor cycles so they are integers; there is at least one instruction z with $\tau(z) = 1$, so the greatest common divisor is also equal 1. The way of estimation of the capacity was suggested by C. Shannon [10], who showed that the capacity $C(I)$ is equal to the logarithm of the largest real solution X_0 of the following characteristic equation:

$$X^{-\tau(z_1)} + X^{-\tau(z_2)} + \cdots + X^{-\tau(z_n)} = 1, \tag{2}$$

where n is the size of I. In [1] it was also proved that the Computer Capacity of multi-core processing unit is defined as the sum of Computer Capacities of the cores. As shown in [1], if a computer contains n computational cores and their Computer Capacities are C_1, C_2, …, C_n, respectively, so the Computer Capacity of such computer will be:

$$C_{\text{comp}} = \sum_{i=1}^{n} C_i. \tag{3}$$

If all cores are the same, i. e. $C_{\text{core}} = C_1 = C_2 = \cdots = C_n$, the formula can be written as $C_{\text{comp}} = n \times C_{\text{core}}$.

Let us show how to calculate the Computer Capacity. Let us describe a simple computer with 8 registers, 16 memory cells of level-1 cache and 256 memory cells of RAM. The list of instructions for this computer consists of: *mov r r*, *mov r m*, *add r r*, *mul r r*. Here, operand r means addressing the register and operand m means addressing the memory cell. The memory in the presented computer is arranged as follows: first, the instruction accesses the cache memory and if the required memory cell is not found there, then it accesses the RAM. Let us consider cache-memory access time as 1 clock cycle and the RAM access time as 5 clock cycles. The execution times of instructions are 1, 1, 2 and 5 cycles in accordance with the list presented above. As mentioned above, the instruction set includes not only the name of the instruction, but also the value of its operands. Also we suppose that all combinations of operands values are possible, so there are $8 \times 8 = 64$ variants of instruction *mov r r*. All of these variants are included in the equation (2) as term $64/X^1$. Similarly the instruction *add r r* is included in the equation as $64/X^2$ and the instruction *mul r r* as $64/X^5$. Consider separately the instruction *mov r m*. When adding this instruction there are two cases: if instruction addresses cache memory and if the memory cell is not found in cache and instruction addresses RAM memory. In the first case there are $8 \times 16 = 128$ different variants with the execution time equals to the sum of execution time of instruction and the cache-memory access time $1 + 1 = 2$, so we include this in equation (2) as term $128/X^2$. In the second case the number of different variants of instruction is $8 \times 256 = 2048$ and the execution time is the sum of instruction execution time, cache-memory access time and RAM access time $1 + 1 + 5 = 7$. So it included in the equation as $2048/X^7$. Now we can construct the equation (2) as:

$$\frac{64}{X^1} + \frac{64}{X^2} + \frac{64}{X^5} + \frac{128}{X^2} + \frac{2048}{X^7} = 1.$$

After using the bisection method we get the value of largest real solution $X_0 = 66.871$ and $C(I) = \log_2 X_0 = 6.06$ bits/clock.

2. The analysis of the characteristic equation

In previous works, we showed that the results of applying the method described above correlate well with the results of generally accepted benchmarks. In the appendix A, one can see graphs with the results of the comparison and links to detailed descriptions of equations for real processors of different architectures. In this section, we show how changes in certain parameters of the processor architecture affect its performance, and how this relates to the method of evaluating the Computer Capacity and, in particular, with the peculiarities of constructing the characteristic equation.

In [8], evolutionary series of Intel processors were investigated, and as a result, it was shown that during the development of Intel processors, manufacturers changed precisely those of the processors architecture complex parameters that affected the Computer Capacity the most. Such parameters are the number of internal registers of the processor, the number of fast instructions and their types. In addition, it was possible to estimate the impact of changes in these parameters quantitatively, and this assessment coincided with practical data. For example, we showed that if the Wolfdale processor increases the number of internal registers 10 times and adds 16 instructions with three register operands whose execution time is 1 clock cycle, the growth of the Computer Capacity will be $\sim$47.5 %. And the next processor in Intel's evolutionary series, Ivy Bridge, was changed exactly the same way, the number of registers increased to 160 from 16 and 10 new instructions of the specific type were added (besides, major changes were made to the list of instructions as a whole). The growth of the Computer Capacity was $\sim$53 %. It is interesting to note that changing other parameters, such as cache and RAM and access time to this memory (of course, changing the access time was considered in real terms, without bringing it to instant access, like registers), had almost no influence on the value of the Computer Capacity. What is important is the fact that, in practice, these values also remained unchanged.

3. Dead-end branch

Let us show in the example how the Computer Capacity could be useful for developers. In this example, it can be argued that the processors development aims to increase the Computer Capacity and if such an increase does not occur it may lead to a dead-end. The processor Pentium III was presented in the beginning of 1999 and its successor Pentium IV was presented in November of 2000. It is important to note that both processors based on architectures with different principles. The main task for a new processor was to increase the maximal clock rate.

We have estimated the values of the Computer Capacity for Pentium III and Pentium IV processors and they were equal to 42.021 and 39.657 bits per clock cycle respectively. This suggests that with equal clock rates the Pentium III processor can perform more tasks than the Pentium IV processor. Moreover, we compared these two processors using the values of Computer Capacity and some popular benchmarks (detailed results are described in the appendix). The most interesting result in this comparison is that all the benchmarks except one show the same results. Clock rate of the investigated Pentium IV processor two times more than Pentium III, but the performance shown on benchmarks turned out to be lower than this value. Another interesting fact is that the average rating of all presented benchmarks close to our characteristic and it is another practical confirmation of its high applicability. Based on this we can assume that the development went the wrong way.

Nevertheless, Pentium IV processors developed and were released for several more years, until they reached the peak of their performance. After that Intel Core architecture appeared, which was based on the Pentium III, and currently, modern Intel processors to a certain extent are the heirs of this architecture. Based on the above, we assume that the mentioned Pentium IV processors can be considered as a dead-end branch of evolution. And we shown how the appearance of such a branch could have been avoided by using in the development the Computer Capacity characteristic which clearly showed the predominance of the first processor over the second.

4. Factors affecting processors architectures

In this section, we highlight and consider the key factors affecting the development of processors. Here we do not consider the factors whose impact on performance is obvious, and focus on the description of factors with not so obvious effect. First we need to list these factors:

- The factors that linearly affect the performance of processor. There is only one such factor, the number of so-called threads. By thread we mean the property of a pipeline to simultaneously execute several unrelated instructions. The number of threads is the maximum number of instructions that the processor pipeline is capable to execute at a time. And it is important to note that such threads use the same memory, i. e. increasing the number of threads does not require an increase in memory.
- The factors with non-linear impact on the performance. These factors include the amounts of different types of memory, the access times of these memory types and the set of instructions. These factors have a tendency to rapid saturation, so it is necessary to investigate, among other things, their growth potential.
- The factor of the little effect of slow instructions on the performance. The concept of the Computer Capacity shows us that the more the instruction execution time, the less its influence on the Computer Capacity, and this influence decreases exponentially. It is important to note that this factor is directly related to the previous one and moreover, follows from it.

4.1. Effect of using shared memory

Formula (3) allow us to understand how the number of cores affects the Computer Capacity. But, in addition, the same formula is applicable to so-called threads. Let us consider this characteristic in more detail. The processor pipeline consists of stages, each of which performs its specific task (instructions decoder, renaming registers, execution blocks, etc.). All stages are characterized by the number of instructions which they can execute in parallel (let us call this the throughput of the stage). By the number of processor threads we will mean the minimum value of throughput among all stages of the pipeline (obviously, it is impossible to execute more instructions in parallel than on the stage with minimum throughput).

Threads are independent sequences of instructions (for example, if there is an instruction which loads data from a memory cell into a register, and the second instruction which adds this register to another one, they cannot be executed in parallel, and therefore they fall into dependency chain) which can be executed in parallel, distribution of instructions on chains of dependencies is made by the processor automatically. In fact, in the context of the Computer Capacity, we can assume that these threads are independent processor cores

with shared memory (such as registers, cache-memory and RAM). As mentioned earlier, we estimate the upper bound of the Computer Capacity, and the set of all pairs of instructions sequences which can be executed in parallel, obviously, is a subset of the set of all possible pairs of instructions sequences. Increasing the number of threads is an extremely profitable solution from the manufacturer's point of view, since all threads inside the kernel use the same registers and memory, but adding a thread affects the performance of the processor, ideally, just like adding a new computational core.

To be more concrete let us look at Haswell processors pipeline. Work [11] provides a detailed description of the pipeline of processors with Haswell microarchitecture. And it is shown in this work that all blocks of the Haswell pipeline can execute simultaneously at least 4 micro-operations. Some blocks can perform more than 4 micro-operations at the same time, but others, limited to just four, will form the so-called "bottlenecks". Therefore, it is necessary to consider exactly 4 possible threads. Thus, ideally, any sequence of instructions can be divided into 4 independent chains of instructions that will be executed simultaneously and will work with same register file, which means that this processor have four independent threads. Since we estimate the upper limit of Computer Capacity, it is possible to assume that the processor always has 4 independent threads.

In part, the influence of this factor can be observed in graphics processors [12]. Despite the fact that the number of computing cores is increased in graphics processors, these cores are inherently close to the threads we described above. For example, the typical NVIDIA GPU comprises of a set of Streaming Multiprocessors (SM) which share the level-2 cache and DRAM. Each SM, in turn, comprises of several Stream Processor (SP) cores which share the level-1 cache memory and the register file. Thus, we can match stream processors and threads because of the fact that both of them, according to our theory, are reduced to computational cores with shared fast memory. And it is interesting especially due to the fact that modern GPUs consist of thousands of SPs.

4.2. Logarithmic saturation effect

The concept of the Computer Capacity allows us to understand why changing different parameters affects the processors performance differently, and, moreover, quantify this effect. First, note that the value of the Computer Capacity is the logarithm of the largest positive root $\log X_0$ of the equation (2). From mathematical analysis it is well known that when the x parameter is changed, the function will change as follows $\ln(x + \Delta x) - \ln x \sim \Delta x/x$. For example, the value of (2) solution for Intel Wolfdale processor equals $X = 13\,007\,226$ and its Computer Capacity value is $C = \log_2 X = \log_2 13\,007\,226 \approx 23.632$. So when we increase ten times the number of vector registers in this processor the value of solution becomes $X = 1\,472\,312\,547$ and the Computer Capacity is $C = \log_2 1\,472\,312\,547 \approx 30.455$ which fully corresponds to the theory described above. It is important to note that the effect of an increase in the number of instructions or memory cells (including the number of registers) also has a non-linear effect on the result of solving the equatio (2). All the above suggests that the considered here non-obvious parameters have a clear tendency to saturation.

Since accessing registers does not imply a delay, an increase in their number will affect almost all the terms, in the denominator of whose exponent is equal to 1 and 2, those that most strongly influence the value of the Computer Capacity. And if in the instruction two operands are registers, then with an increase in the number of registers 2 times, the numerator will increase 4 times, and if there are three operands, then, respectively, 8 times. In [8], it

was shown that with a tenfold increase in the number of integer and vector registers, the value of the Computer Capacity of Intel processors increases by more than 30 %. The same increase in the number of internal registers can be observed in the transition from Wolfdale processors to Ivy Bridge processors. Unfortunately, to get a similar growth, developers need to increase the current number of registers 10 times, but in the Skylake architecture there are already 180 integer and 168 vector internal registers. It will be problematic to achieve this with the existing element base, so we can talk about the approaching saturation.

One of the most difficult factors to evaluate is the number of instructions. There are several ways, the development of new instructions with a large number of operands (as, for example, it was done in Intel processors, when fast instructions with three register operands were added), or the addition of new instructions of the existing type. For example, ARM followed the path of a significant increase in the number of instructions, which can be seen by comparing the architecture of the Cortex-M3 and Cortex-A57 processors. In [13] you can find a detailed description of all architectures of ARM processors, and the link [14] contains transformed instruction sets, on the basis of which we built the characteristic equation (2) for these processors. In these converted lists (they are presented in a special format, where each instruction is divided into terms depending on its execution type), the description of the processor M3 contains 218 instructions, while the list for A57 consists of 1877 instructions. If you look at the equations, you can see that the numerator in the first term (where the degree is 1) in A57 equation (4) is three times greater than the numerator of the first term in M3 equation (5). At the same time, the Computer Capacity of Cortex-M3 $C(I_{M3}) = 25.2$ bits/clock increased to $C(I_{A57}) = 29.79$ bits/clock for Cortex-A57, i. e. only 18 %. Nevertheless, the A57 processor significantly exceeds the performance of the M3, primarily due to the fact that the number of threads was increased from 1 to 3, which allowed it to get closer to modern Intel processors (comparing with Intel processors using benchmarks and computing capacity can be found in the appendix). In addition, the development of new instructions is a very time-consuming process, so in this direction we can also note the approaching saturation.

$$\frac{928619266}{X^1}+\frac{358487379}{X^2}+\frac{72121984}{X^3}+\frac{279500883}{X^4}+\frac{167809692}{X^5}+\frac{31454404}{X^6}+ \\ +\frac{3735614}{X^7}+\frac{25589699}{X^8}+\frac{21567538}{X^9}+\frac{7511822}{X^{10}}+\frac{102413}{X^{11}}+\frac{131072}{X^{12}}+ \\ +\frac{65536}{X^{14}}+\frac{65536}{X^{15}}=1, \qquad (4)$$

$$\frac{38553518}{X^1}+\frac{143623384}{X^2}+\frac{35360612}{X^3}+\frac{35568868}{X^4}+\frac{2142526}{X^5}+\frac{1321936}{X^6}+ \\ +\frac{2239104}{X^7}+\frac{2922720}{X^8}+\frac{3094292}{X^9}+\frac{2579488}{X^{10}}+\frac{1687168}{X^{11}}+\frac{857248}{X^{12}}+\frac{327600}{X^{13}}+ \\ +\frac{92400}{X^{14}}+\frac{18000}{X^{15}}+\frac{2160}{X^{16}}+\frac{120}{X^{17}}=1. \qquad (5)$$

4.3. Low effect of slow instructions

As was shown, the amount of memory affects the value of the numerators in the addends, which are formed by instructions working with memory. Let there be an instruction *instrm* whose execution time is 1 clock cycle, while the computer has a 1-level cache memory with

size L_1, a cache memory of the 2nd level with size L_2 and RAM with size M memory cells. The memory access time is denoted by $\tau(L_1)$, $\tau(L_2)$, $\tau(M)$, respectively. Then such an instruction forms three components: $\frac{L_1}{X^{1+\tau(L_1)}}$, $\frac{L_2}{X^{1+\tau(L_1)+\tau(L_2)}}$, $\frac{M}{X^{1+\tau(L_1)+\tau(L_2)+\tau(M)}}$. If we increase the cache size of the first level 2 times, the numerator in the first term becomes $2L_1/X^{1+\tau(L_1)}$. However, it is important to note here that, for example, in Intel processors, the access time to the 1-level cache is 3–4 clocks, and therefore the denominator will have a 4–5 exponent value. It was shown in [8] that the processor has a sufficient number of instructions forming terms with exponent 1 or 2, and they have a much greater influence on the solution of the equation (2), and, therefore, even a tenfold increase in 1-level cache memory practically does not affect the final value of the Computer Capacity. In [8], we showed that, for Intel processors, a change in the cache sizes of all levels, as well as a change in the size of RAM, does not affect the Computer Capacity, and therefore does not make much sense to the performance. Let us look for example at Ivy Bridge equation (6).nThe solution of this equation $X \approx 94\,472\,124.2$ and $C(I) \approx 26.493$. And in this equation all the instructions with memory operands have exponent value 5 and greater. But if we drop all the instructions with memory operands from equation and try to evaluate the Computer Capacity, the achieved result changes minimally. We do not claim that these instructions are not needed, but this example shows that they have almost no effect on performance value. And, as practice has shown, manufacturers really did not change the volume of these types of memory, that is, saturation was achieved. Such a trend could have been significantly reduced by the access time to this memory, however, unfortunately, the technologies used in the investigated processors do not allow this to be done yet. We need to note that such effect of slow instructions corresponds to logarithmic factor, and, in fact, well illustrates the essence of the saturation effect.

$$\frac{94472115}{X^1}+\frac{870080273}{X^2}\frac{18320178}{X^3}+\frac{3078979}{X^4}+\frac{1110704132}{X^5}+\frac{1249083393}{X^6}+\frac{49703166209}{X^7}+$$
$$+\frac{523714076}{X^8}+\frac{3973281}{X^9}+\frac{905632}{X^{10}}+\frac{2649625}{X^{11}}+\frac{2490370}{X^{12}}+\frac{1310880}{X^{13}}+\frac{1310720}{X^{14}}+$$
$$+\frac{5144576}{X^{15}}+\frac{1}{X^{16}}+\frac{4442816514}{X^{17}}+\frac{4997665025}{X^{18}}+\frac{198813974529}{X^{19}}+\frac{2094687848}{X^{20}}+\frac{15892482}{X^{21}}+$$
$$+\frac{6029312}{X^{22}}+\frac{10511975}{X^{23}}+\frac{9961472}{X^{24}}+\frac{5242881}{X^{25}}+\frac{6553601}{X^{26}}+\frac{20578304}{X^{27}}+\frac{1310722}{X^{28}}+$$
$$+\frac{5242880}{X^{30}}+\frac{5242881}{X^{31}}+\frac{9961472}{X^{34}}+\frac{161}{X^{35}}+\frac{1}{X^{37}}+\frac{5242880}{X^{38}}+\frac{5242880}{X^{40}}+\frac{1}{X^{41}}+$$
$$+\frac{1}{X^{42}}+\frac{533137987993}{X^{47}}+\frac{599560028160}{X^{48}}+\frac{23857519656960}{X^{49}}+\frac{251362541568}{X^{50}}+\frac{1907097600}{X^{51}}+$$
$$+\frac{424673281}{X^{52}}+\frac{1261436928}{X^{53}}+\frac{1195376640}{X^{54}}+\frac{629145600}{X^{55}}+\frac{629145601}{X^{56}}+\frac{2469396480}{X^{57}}+$$
$$+\frac{26374}{X^{59}}+\frac{629145600}{X^{60}}+\frac{629145600}{X^{61}}+\frac{1195376640}{X^{64}}+\frac{629145600}{X^{68}}+\frac{629145600}{X^{70}}+$$
$$+\frac{291164422930432}{X^{77}}+\frac{327439716712448}{X^{78}}+\frac{13029386735321088}{X^{79}}+\frac{137277462701670}{X^{80}}+$$
$$+\frac{1041529569280}{X^{81}}+\frac{231928233984}{X^{82}}+\frac{688912754278}{X^{83}}+\frac{652835028992}{X^{84}}+\frac{343597383680}{X^{85}}+$$
$$+\frac{343597383680}{X^{86}}+\frac{1348619730944}{X^{87}}+\frac{3145728}{X^{89}}+\frac{343597383680}{X^{90}}+\frac{343597383680}{X^{91}}+$$

$$+\frac{652835037184}{X^{94}}+\frac{343597383680}{X^{98}}+\frac{343597383680}{X^{100}}+\frac{1}{X^{102}}+\frac{32768}{X^{106}}+\frac{1717986918}{X^{119}}+\frac{128}{X^{120}}+$$
$$+\frac{512}{X^{132}}+\frac{128}{X^{134}}+\frac{3932160}{X^{136}}+\frac{512}{X^{146}}+\frac{8192}{X^{147}}+\frac{32768}{X^{159}}+\frac{61440}{X^{162}}+\frac{2147483648}{X^{166}}+$$
$$+\frac{61440}{X^{176}}+\frac{3932160}{X^{189}}+\frac{33554432}{X^{192}}+\frac{33554432}{X^{206}}+\frac{2147483648}{X^{219}}+$$
$$+\frac{6553}{X^{247}}+\frac{26214}{X^{259}}+\frac{3145728}{X^{289}}+\frac{1717986918}{X^{319}}=1. \quad (6)$$

Conclusion

We have shown how the concept of the Computer Capacity helps us to identify and, moreover, to quantify the factors which influence the development of processors. In this paper, we have identified the factors in computer development whose influence on the performance is not so obvious. We divided these factors into two groups, with linear impact on the performance and with the impact close to logarithmic. Most of the parameters of the second group have already reached saturation, but among them are those who have not reached the growth limit. These parameters differ for different processors, for example, ARM processors have potential to grow the number of registers, while Intel processors are already close to the limit of this value (it will be quite problematic to increase their number tenfold). In its turn, the factors from the first group most evident in GPUs, but in fact they are used in all modern processors. It is important to note that the change in the element base of the processors also agrees well with the presented theory. For example, increasing the speed of access to memory will affect the denominators in the characteristic equation and will undoubtedly increase the Computer Capacity, and we have shown how to estimate this increase.

In previous years, manufacturers implicitly took these factors into account, but unconsciously, based on experiments, benchmarks and their own intuition. It seems to us that explicit consideration of the Computer Capacity and corresponding equation (2) as well as described factors can significantly effect the development of processors. Here we can give some conditional analogy with the "evolution" of optical devices. Obviously, the "evolution" did not know what the focal length, refractive index and other characteristics of the optical systems. In the animal world, vision organs developed unconsciously obeying the main optic characteristics (such as the focal length of the lens, which is essentially the basis of the eye) and the laws of optics, but after the advent of optical devices, their development took place with an explicit account of these laws, which, it seems to us, has greatly accelerated this process. We hope that the laws described above, as well as the concept of the Computer Capacity in general, can accelerate the development of processors, as well as devices in which they are used.

Acknowledgements. This research was funded by the Russian Foundation for Basic Research (grant No. 18-29-03005).

Appendix A

In Fig. 1 we present the results of comparison Pentium III and Pentium IV processors using Computer Capacity and the values of popular benchmarks [4, 15–17]. We compared Intel Pentium IV Processor 2.60 GHz and Intel Pentium III 1.26 GHz-S and results presented in

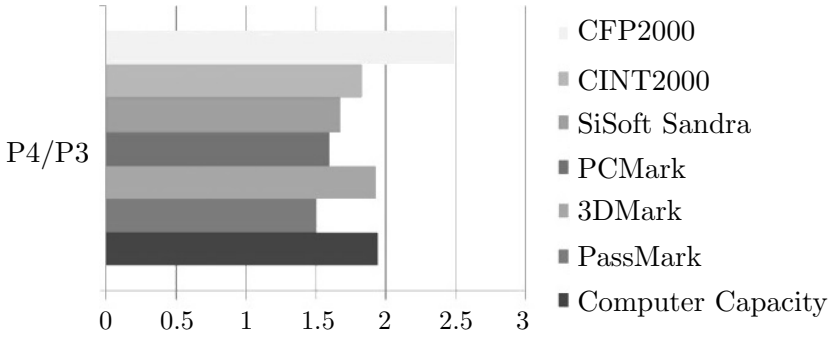

Fig. 1. The results of Pentium III and Pentium IV comparison

T a b l e 1. The results of Pentium III and Pentium IV benchmarks and Computer Capacity values

Processor	Frequency, MHz	Computer Capacity, Mbits/s	PassMark	3DMark	PCMark	SiSoft Sandra	CINT 2000	CFP 2000
Pentium III	1266	53 200.28	270	1375	1922	4150	564	422
Pentium IV	2600	103 115.51	405	2652	3070	6955	1030	1052

relative form by dividing the value of Pentium IV (P4) result by Pentium III (P3) result. The values of benchmarks and Computer Capacity presented in Table 1.

Appendix B

Here we present some results of comparing the Computer Capacity with benchmarks for processors with different architectures. In Table 2 we present the raw values of benchmarks and Computer Capacity. Considering the difference in units of measurement, it is not possible to directly compare our characteristic with benchmarks, so a relative comparison has been made. Fig. 2 shows how the performance of different Intel processors varies relatively to the ARM Cortex-A57 processor. The comparison presented for single core of each processor, the benchmarks results are taken from [18]. All the characteristic equations and calculation programs can be found in [14].

T a b l e 2. The results of Intel processors and ARM Cortex-A57 benchmarks and Computer Capacity values

Processor	kDhry	Whet	Execl	kCopy	kPipe	Index	Computer Capacity, Mbits/s	Frequency, MHz
SkyLake	36 998	4572.5	5271.4	941.08	1819.1	1708.2	371 866.394	3200
Haswell	31 253	6588.3	3598.5	533.37	836.6	1093.7	428 682.474	3700
Ivy Bridge	30 114	3559.4	4196.9	684.29	1447.9	1324.4	286 128.538	2700
Cortex-A57	18 548	2058.5	1993.9	217.69	373.4	498.6	178 743.072	2000

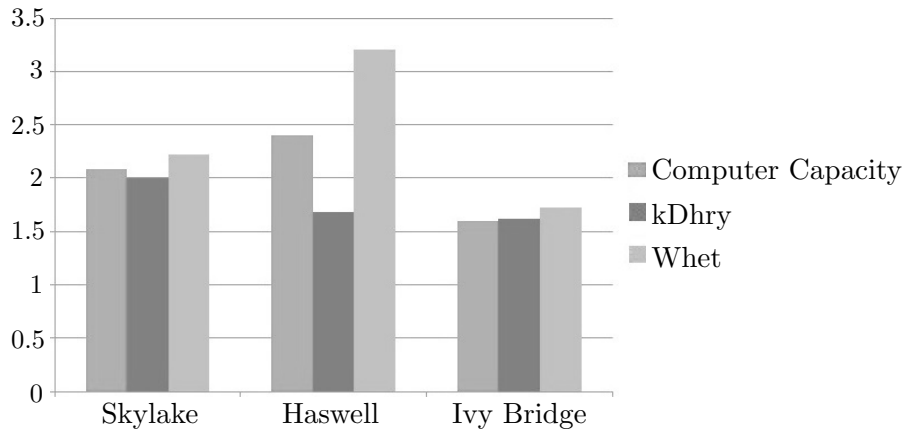

Fig. 2. The results of processors comparison

References

[1] **Ryabko B.** An information-theoretic approach to estimate the capacity of processing units. Performance Evaluation. 2012; 69:267–273.

[2] **Ryabko B., Rakitskiy A.** An analytic method for estimating the computation capacity of computing devices. Journal of Circuits, Systems and Computers. 2017; 26(05):1750086.

[3] **Ryabko B., Rakitskiy A.** Theoretical approach to performance evaluation of supercomputers. Journal of Circuits, Systems and Computers. 2018; 27(04):1850062.

[4] PassMark Software, Passmark performance test description. 2019. Available at: `http://www.passmark.com/support/performancetest/interpreting_test_results.htm` (Accessed May 28, 2019).

[5] **Gal-On S., Levy M.** Exploring coremark a benchmark maximizing simplicity and efficacy. The Embedded Microprocessor Benchmark Consortium. 2012. Available at: `https://www.eembc.org/techlit/articles/coremark-whitepaper.pdf` (Accessed May 28, 2019).

[6] The list of top 500 supercomputers. Call for participation in the 56th TOP500/GREEN500 lists. 2019. Available at: `https://www.top500.org` (Accessed May 28, 2019).

[7] **Lilja D.J.** Measuring computer performance: A practitioner's guide. Cambridge: Cambridge Univ. Press; 2005.

[8] **Ryabko B., Rakitskiy A.** Investigation of the processors evolution using the computer capacity. International Symposium on Information Theory and Its Applications (ISITA). IEEE, 2018; 404–408.

[9] **Rakitskiy A., Ryabko B.** Computer Capacity as a tool for the processors development analysis. XVI International Symposium "Problems of Redundancy in Information and Control Systems" (REDUNDANCY). Moscow, Russia, 2019; 186–190.

[10] **Shannon C.E.** A mathematical theory of communication. Bell System Technical Journal. 1948; 27(3):379–423.

[11] **Kanter D.** Intel's Haswell CPU microarchitecture. Real World Technologies. 2012: 17.

[12] **Nickolls J., Dally W.J.** The GPU computing era. IEEE Micro. 2010; 30(2):56–69. DOI:10.1109/MM.2010.41.

[13] ARM, all technical manuals and other documentation for arm products. 2019. Available at: `https://developer.arm.com/docs` (Accessed May 28, 2019).

[14] **Rakitskiy A.** Laboratory of information systems and data protection. Available at: `http://www.ict.nsc.ru/ru/structure/orgunits/lab-info-sys-security-page` (Accessed May 28, 2019).

[15] Standard performance evaluation corporation, SPEC CPU2000 V1.3. Available at: `https://www.spec.org/cpu2000` (Accessed March 30, 2020).

[16] SiSoftware Sandra processor benchmark. Available at: `https://www.sisoftware.co.uk` (Accessed March 30, 2020).

[17] UL Benchmarks, PCMark and 3DMark benchmarks. Available at: `https://benchmarks.ul.com` (Accessed March 30, 2020).

[18] O.S.A.D. Lab, Single-core CPU unixbench benchmark. 2019. Available at: `https://www.osadl.org/CPU-benchmarks.qa-farm-cpu-benchmarks.0.html` (Accessed May 28, 2019).

Вычислительные технологии, 2020, том 25, № 6, с. 104–115. © ФИЦ ИВТ, 2020 ISSN 1560-7534
Computational Technologies, 2020, vol. 25, no. 6, pp. 104–115. © FRC ICT, 2020 eISSN 2313-691X

INFORMATION TECHNOLOGIES

DOI:10.25743/ICT.2020.25.6.007

Теория информации как инструмент для определения основных факторов, влияющих на архитектуру процессоров

А. А. Ракитский[1,2,3,*], Б. Я. Рябко[1,2]

[1] Новосибирский государственный университет, 630090, Новосибирск, Россия

[2] Федеральный исследовательский центр информационных и вычислительных технологий, 630090, Новосибирск, Россия

[3] Сибирский государственный университет телекоммуникаций и информатики, 630009, Новосибирск, Россия

[*] Контактный автор: Ракитский Антон Андреевич, e-mail: `apple-66@mail.ru`

Поступила 15 июня 2020 г., доработана 22 октября 2020 г., принята в печать 30 октября 2020 г.

Аннотация

В работе исследуется процесс разработки компьютеров за последние десятилетия с целью определения наиболее влияющих на него факторов. Описываются сами факторы, которые используются для предсказания направления будущих разработок. Для решения этой задачи применяется концепция Вычислительной Способности, которая позволяет оценить производительность компьютеров теоретически, опираясь исключительно на описание их архитектуры.

Ключевые слова: Вычислительная Способность, архитектура процессоров, теория информации, характеристики производительности процессоров, теория Шеннона.

Цитирование: Rakitskiy A.A., Ryabko B.Ya. Information theory as a means of determining the main factors affecting the processors architecture. Computational Technologies. 2020; 25(6): 104–115. DOI:10.25743/ICT.2020.25.6.007.

Благодарности. Исследование выполнено при поддержке РФФИ (грант № 18-29-03005).