

# *Two Centuries of Productivity Growth in Computing*

WILLIAM D. NORDHAUS

The present study analyzes computer performance over the last century and a half. Three results stand out. First, there has been a phenomenal increase in computer power over the twentieth century. Depending upon the standard used, computer performance has improved since manual computing by a factor between 1.7 trillion and 76 trillion. Second, there was a major break in the trend around World War II. Third, this study develops estimates of the growth in computer power relying on performance rather than components; the price declines using performance-based measures are markedly larger than those reported in the official statistics.

The history of technological change in computing has been the subject of intensive research over the last five decades. However, little attention has been paid to comparing the performance of modern computers to pre-World War II technologies or even pencil-and-pad calculations. The present study investigates the progress of computing over the last century and a half, including estimates of the progress relative to manual calculations.<sup>1</sup>

The usual way to examine technological progress in computing is either through estimating the rate of total or partial factor productivity or through examining trends in quality-adjusted prices. For such measures, it is critical to use constant-quality prices so that improvements in the capabilities of computers are adequately captured. The earliest studies, dating from around 1953, examined the price declines of mainframe computers and used computers. Early studies found annual price declines of 15 to 30 percent per year, and recent estimates find annual price declines of 25 to 45 percent.<sup>2</sup>

*The Journal of Economic History*, Vol. 67, No. 1 (March 2007). © The Economic History Association. All rights reserved. ISSN 0022-0507.

William D. Nordhaus is Sterling Professor of Economics, Department of Economics, Yale University, 28 Hillhouse Avenue, New Haven, CT 06520; Professor, School of Forestry and Environmental Studies, Yale University; and Research Associate, NBER. E-mail: william.nordhaus@yale.edu.

This is a revised version of an earlier paper entitled, "The Progress of Computing." The author is grateful for comments from Ernst Berndt, Tim Bresnahan, Paul Chwelos, Iain Cockburn, Ray Fair, Robert Gordon, Steve Landefeld, Phil Long, Chuck Powell, and three anonymous referees. Eric Weese provided extraordinary help with research on computer history.

<sup>1</sup> The data used in this study are provided in a background spreadsheet available at <http://www.econ.yale.edu/~nordhaus/Computers/Appendix.xls>. That spreadsheet contains 14 pages with details of the calculations used in this article. The contents of the spreadsheet are in the page labeled "Contents." The relevant page of the spreadsheet is cited here as "Nordhaus, *Online Appendix*, page x," when referring to page x of this source as documentation.

<sup>2</sup> Table 10 provides some documentation. See Landefeld and Grimm, "Note," pp. 17–22, for a discussion and a compilation of studies.

Although many analysts are today examining the impact of the “new economy” and especially the impact of computers on real output, inflation, and productivity, one might naturally wonder how new the new economy really is. Mainframe computers were crunching numbers long before the new economy appeared on the radar screen, and mechanical calculators produced improvements in computational capabilities even before that. How does the progress of computing in recent years compare with that of earlier epochs of the computer and calculator age? This is the question addressed in the current study.

#### A SHORT HISTORY OF COMPUTING

We begin with some fundamentals of computational theory. Abacuses, manual arithmetic, calculators, and computers are information-processing systems. They involve taking an array of data (binary digits, words, and so on) and transforming them according to some functional relationship or algorithm (such as adding, simulating, solving a differential equation, or producing a PET image) into a new array. Technological change in computing is the development of new software, hardware, communications, and systems that can expand the range of problems that can be solved using computational techniques and the speed at which they are solved.

Computers are such a pervasive feature of modern life that we can easily forget how much of human history existed with only the most rudimentary aids to addition, data storage, printing, copying, rapid communications, or graphics. The earliest recorded computational device was the abacus, but its origins are not known. The Darius Vase in Naples (dated around 450 BC) shows a Greek treasurer using a table abacus, or counting board, on which counters were moved to record numbers and perform addition and subtraction. The earliest extant “calculator” is the Babylonian Salamis tablet (300 BC), a huge piece of marble, which used the Greek number system and probably deployed stone counters. Analog devices developed during the first century BC, such as the Antikythera Mechanism, may have been used to calculate astronomical dates and cycles.<sup>3</sup>

The design for the modern abacus appears to have its roots in the Roman hand-abacus, introducing grooves to move the counters, of which there are a few surviving examples. Counting boards looking much like the modern abacus were widely used as mechanical aids in Europe from Roman times until the Napoleonic era, after which most

<sup>3</sup> Freeth, Bitsakis, Moussas, et al., “Decoding the Ancient Greek Astronomical Calculator,” pp. 587–91.

reckoning was done manually using the Hindu-Arabic number system. The earliest records of the modern rod abacus date from the thirteenth century in China (the *suan-pan*), and the Japanese variant (the modern *soroban*) came into widespread use in Japan in the nineteenth century.

Improving the technology for calculations naturally appealed to mathematically inclined inventors. Around 1502 Leonardo sketched a mechanical adding machine; it was never built and probably would not have worked. The first surviving machine was built by Pascal in 1642, using interlocking wheels. I estimate that fewer than 100 operable calculating machines were built before 1800.<sup>4</sup>

Early calculators were “dumb” machines that essentially relied on incrementation of digits. An important step in the development of modern computers was mechanical representation of logical steps. The first commercially practical information-processing machine was the Jacquard loom, developed in 1804. This machine used interchangeable punched cards that controlled the weaving and allowed a large variety of patterns to be produced automatically. This invention was part of the inspiration of Charles Babbage, who developed one of the great precursor inventions in computation. He designed two major conceptual breakthroughs, the “Difference Engine” and the “Analytical Engine.” The latter sketched the first programmable digital computer. Neither of the Babbage machines was constructed during his lifetime. An attempt in the 1990s by the British Museum to build the simpler Difference Engine using early-nineteenth-century technologies failed to perform its designed tasks.<sup>5</sup>

The first calculator to enjoy large sales was the “arithmometer,” designed and built by Thomas de Colmar, patented in 1820. This device used levers rather than keys to enter numbers, slowing data entry. It could perform all four arithmetic operations, although the techniques are today somewhat mysterious.<sup>6</sup> The device was as big as an upright piano, unwieldy, and used largely for number crunching by insurance companies and scientists.<sup>7</sup> Contemporaneous records indicate that 500 were produced by 1865, so although it is often called a “commercial success,” it was probably unprofitable.<sup>8</sup>

Table 1 shows an estimate of the cumulative production of computational devices (excluding abacuses and counting boards) through 1920.

<sup>4</sup> Nordhaus, *Online Appendix*, page “Quant\_History.”

<sup>5</sup> See Swade, *Difference Engine*.

<sup>6</sup> An excellent short biography of this device is available in Johnston, “Making the Arithmometer,” pp. 12–21.

<sup>7</sup> The present author attempted to use a variant of the arithmometer but gave up after an hour when failing to perform a single addition.

<sup>8</sup> Nordhaus, *Online Appendix*, page “Quant\_History.”

TABLE 1  
 PRODUCTION OF COMPUTATIONAL DEVICES UP TO 1920

Period	Production	Cumulative Production
Up to 1800	53	53
1801–1850	57	110
1851–1860	50	160
1861–1870	170	330
1871–1880	370	690
1881–1890	821	1,500
1891–1900	6,700	8,200
1901–1910	120,000	130,000
1911–1920	780,000	910,000

*Note:* Figures are rounded to two significant digits.

*Source:* Nordhaus, *Online Appendix*, page “Table 1.”

This tabulation indicates that fewer than 1,000 mechanical calculators were extant at the time of rise of the calculator industry in the 1870s, so most calculations at that time were clearly done manually.<sup>9</sup> By the 1880s, industrial practice plus the increasing need for accurate and rapid bookkeeping combined to give the necessary impetus for the development of workable commercial adding machines and calculating machines. James Cortada emphasizes the development of the technology underlying the typewriter as a key engineering element in calculator design.

Two different sets of designs were the circular machine and the keyboard design. The circular calculator was designed by Frank Baldwin in the United States and T. Odhner in Russia, both first built in the 1872–1874 period. The second and ultimately most successful early calculator was invented by Dorr E. Felt (1884) and William S. Burroughs (1885). These machines used the now-familiar matrix array of keys, and were produced by firms such as Felt Comptometer, American Arithmometer, Monroe, and Burroughs. Production and sales of calculators began to ramp up sharply in the 1890s, as Table 1 indicates.

It is difficult to imagine the tedium of office work in the late nineteenth century. According to John Coleman, president of Burroughs, “Bookkeeping, before the advent of the adding machine, was not an occupation for the flagging spirit or the wandering mind . . . . It required in extraordinary degree a capacity for sustained concentration, attention to detail, and a passion for accuracy.”<sup>10</sup>

Calculator manufacturers recognized that sales would depend upon the new machines being both quicker and more accurate than early devices

<sup>9</sup> A comprehensive economic history of calculation before the electronic age is presented in Cortada, *Before the Computer*.

<sup>10</sup> Quoted in Cortada, *Before the Computer*, p. 26.

157.38	157.38
762.91	762.91
435.75	435.75
800.76	800.76
43.02	43.02
987.25	987.25
500.00	500.00
1003.50	1,003.50
245.65	245.65
82.47	82.47
4250.86	4,250.86
1014.75	1,014.75
243.92	243.92
914.75	914.75
5475.80	5,475.80
14850.07	14,850.07
0418.25	418.25
.74	.74
9.10	9.10
27.72	27.72
896.35	896.35
1238.63	1,238.63
7800.00	7,800.00
10000.00	10,000.00
127.34	127.34
77.01	77.01
3032.4	3,032.4
3808.89	3,808.89
458.92	458.92
1456789.34	1,456,789.34
11025.22	11,025.22
600.10	600.10
2250.85	2,250.85
14623.45	14,623.45
8207.12	8,207.12
2100.00	2,100.00
<u>15527231.14</u>	<u>15,527,231.14</u>
2	3

Facsimile of Figures Added and Listed by a Clerk in 9 Minutes.

Facsimile of Figures Added and Listed by a "Burroughs" in 1 Min. 30 Sec.

FIGURE 1  
COMPARISON OF RESULTS OF ADDING MACHINE AND CLERK

Note: This plate shows a comparison of manual calculators and computations by a clerk in adding up a column of numbers such as might be found in a ledger. The calculator has an advantage of a factor of six.

Source: Burroughs Adding Machine Company, *Better Day's Work*, pp. 153–54.

or humans, but comparative studies of different devices are rare. A 1909 report from Burroughs compared the speed of trained clerks adding up long columns of numbers by hand with that of a Burroughs calculator, as shown in Figure 1. These showed that the calculator had an advantage of about a factor of six, as reported: Ex-President Eliot of Harvard hit the nail squarely on the head when he said, "A man ought not to be employed at a task which a machine can perform."

Put an eight dollar a week clerk at listing and adding figures, and the left hand column (see Figure 1) is a fair example of what he would produce in nine minutes if he was earning his money.

The column on the right shows what the same clerk could do in one-sixth the time, or one and a half minutes.<sup>11</sup>

<sup>11</sup> Burroughs Adding Machine Company, *Better Day's Work*, pp. 153–54.

TABLE 2  
THE SEVEN STAGES OF COMPUTATION

- 
1. Manual (see Figure 1) — up to around 1900
  2. Mechanical — circa 1623 to 1945
  3. Electromechanical — 1902 to 1950
  4. Relays — 1939 to 1944
  5. Vacuum tubes — 1942 to 1961
  6. Transistor — 1956 to 1979
  7. Microprocessor — 1971 to present
- 

*Note:* The dates are for the technologies in this study.

The early calculators were not well designed for mass data input and output. This problem was solved with the introduction of punched-card technology, adapted circuitously from the Jacquard power loom. The Electrical Tabulating System, designed by Herman Hollerith in the late 1880s, saw limited use in hospitals and the War Department, but its first serious deployment was for the 1890 census. The Tabulator was unable to subtract, multiply, or divide, and its addition was limited to simple incrementation. Its only function was to count the number of individuals in specified categories, but for this sole function, it was far speedier than all other available methods. During a government test in 1889, the tabulator processed 10,491 cards in five and a half hours, averaging 0.53 cards per second.

Over the next half-century, several approaches were taken to improving the speed and accuracy of computation, and the tales of mechanical and electrical engineering have been retold many times. The major technologies underlying the computers examined here are shown in Table 2. Some of the major technological milestones were the development of the principles of computer architecture and software by John von Neumann (1945), the first electronic automatic computer (the ENIAC in 1946), the invention of the transistor (1947) and its introduction into computers (1953–1956), the first high-level programming language (Fortran, 1954), the development of the first microprocessor (1971), personal computers (dated variously from the Simon in 1950 to the Apple I in 1976 or the IBM PC in 1981), the first edition of Microsoft Windows (1983), and the introduction of the world wide web (1989).

Although the engineering of calculators and computers is a much-told tale, virtually nothing has been written on the economics of early calculating devices. The economics of the computer begins with a study by Gregory Chow.<sup>12</sup> He estimated the change in computer prices using three variables (multiplication time, memory size, and access time) to

<sup>12</sup> Chow, “Technological Change,” pp. 1117–30.

measure the performance of different systems over the period 1955–1965. Many studies have followed in this tradition, and Jack Triplett provides an excellent recent overview of different techniques.<sup>13</sup> Overall, there are 253 computing devices in this study for which minimal price and performance characteristics could be identified. The full set of machines and their major parameters are provided in an appendix available online.<sup>14</sup>

## MEASURING COMPUTER PERFORMANCE

### *The Scope and Definition of Computer Power*

The present study focuses on the long-term trend in the prices and productivity of “computer power.” It will be useful to begin with a definition of this term and an explanation of its scope and limitations. The central measure of computer power is the rate at which calculators or computers could execute certain standard tasks, measured in computations per second. This measure has an important advantage in comparison to most other measures of “real output” because it can be measured directly rather than by taking dollar values and applying price deflators. Moreover, because scientists and engineers are particularly interested in computer performance, there have been careful measurements of these data for over half a century.

In constructing the measures of computer power, I rely upon data on the costs and performance data of different machines over the last two centuries, with particular focus on the last hundred years. These data involve costs and inputs of capital and labor, which are relatively straightforward to obtain; additionally, I estimate performance in terms of time to perform standardized computational tasks, which turn out to be much more difficult to measure.

The bundle of computations performed by different systems evolves greatly over time. For the earliest calculators, the tasks involved primarily addition (say for accounting ledgers). To these early tasks were soon added scientific and military applications (such as calculating ballistic trajectories, design of atomic weapons, and weather forecasts). In the modern era, computers are virtually everywhere, making complex calculations in science and industry, helping consumers e-mail or surf the web, operating drones on the battlefield, producing images from medical scans, and combating electronic diseases. In all cases, I measure “computer power” as the number of times that a given bundle of

<sup>13</sup> Triplett, “Performance Measures,” pp. 97–140.

<sup>14</sup> Nordhaus, *Online Appendix*, page “Data.”

TABLE 3  
ALTERNATIVE MEASURES OF PERFORMANCE USED IN THIS STUDY

---



---

1. Addition time (up to about 1944)
2. Millions of instructions or operations per second (1944 to 1980s)
3. Moravec's formula: Performance a function of (add-time, multiplication-time, memory, word size)
4. Knight's formula (1944 to 1972) Performance a function of (word size, memory, calculation-time, IO time, ...)
5. Synthetic benchmarks: Dhrystone (1984 to 1990) WorldBench SYSmark SPEC (latest being SPEC2000): 1993 to present

---

*Notes:* Benchmarks used in measuring computer performance have evolved from the speed of addition or multiplication to the performance on a suite of complex tasks. The tasks used in the latest SPEC benchmark are shown in Table 4.

computations can be performed in a given time; and the cost of computation as the cost of performing the benchmark tasks.

An *ideal measure of computer performance* would follow the principles of standard index number theory. For example, it would take an evolving mix of tasks  $\{X_1(t), \dots, X_n(t)\}$  along with the prices or costs of these tasks  $\{P_1(t), \dots, P_n(t)\}$ . The tasks might be {addition, subtraction, multiplication, . . . , flight simulation, Internet access, playing chess, . . . , DNA sequencing, solving problems in quantum chromodynamics, . . .}. The prices would be the constant-quality prices of each of these activities (using the reservation price when the activity level is zero). In principle, we could use Törnqvist or other superlative indexes to construct chained cost indexes.

In practice, construction of an ideal measure is far beyond what is feasible with existing data. The first shortcoming is that there is virtually no information on either the mix or relative importance of applications over time or on the market or implicit prices of different applications. Measuring computer power has bedeviled analysts because computer characteristics are multidimensional and evolve rapidly over time. The absence of reliable data on performance has forced economic studies of computer prices (called “hedonic” pricing studies) to draw instead on the prices of the input components of computers. The hedonic approach is not taken in this study but will be discussed in a later section.

As a substitute for the ideal measure, the present study has linked together price measures using changing bundles of computational tasks. The tasks examined here have evolved over time as the capabilities of computers grew. Table 3 gives an overview of the different measures of performance that are applied to the different computers. This approach

means that the indexes largely involve addition and multiplication speed for the early years but involve the speed for complex procedures for the later years.

A second major shortcoming is that the present study does not account adequately for the contribution of complementary inputs into computational technologies. In early years, where the devices were simple adding machines, the major complementary factor was a roll of paper, and the omission is probably minor. In the modern era, software, high-level languages, and input-output technologies were complementary components of the production process. More recently, high-speed data transmission, video capabilities, multitasking, and Internet connectivity have been essential parts of computational capacity. The present study does not include either the costs or the productivity of these complementary technologies, but they have clearly been an important part of the rapid growth in both the speed and breadth of productivity growth.

#### *Details on Measures of Computer Performance*

This section describes the measures of computer performance that have been used to construct the time series. I begin with an overview and then describe the procedures in detail. The purpose of these measures is to develop a time series of computer performance from the earliest days to the present. I designate *CPS* or “computations per second” as the index of computer power, and *MCPS* as “millions of computations per second.”

For ease of understanding, I have set this index so that the speed of manual computations equals one. As a rough guide, if you can add two five-digit numbers in seven second and multiply two five-digit numbers in 80 seconds, you have one unit of computer power. The earliest devices, such as counting boards, the abacus, and adding machines, were primarily designed for addition; these could sometimes parlay addition into other arithmetic functions (multiplication as repeated addition). The earliest metric of computer performance therefore is simply addition time. This is converted into a measure of performance that can be compared with later computers using alternative benchmark tests. For computers from around World War II until around 1975, we use a measure of performance developed by Kenneth Knight that incorporates additional attributes. For the modern period, we use computer benchmarks that have been devised by computer scientists to measure performance on today’s demanding tasks.

## ADDITION TIME

The earliest devices were adding machines. Figure 1 shows the results of a typical task as described in 1909. In fact, until World War II, virtually all commercial machines were devoted solely to addition. We can compare the addition time of different machines quite easily as long as we are careful to ensure that the length of the word is kept constant for different machines.

## MORAVEC'S INFORMATION-THEORETIC MEASURE OF PERFORMANCE

A measure of performance that relies primarily on arithmetic operations but has a stronger conceptual basis is the information-theoretic measure devised by Hans Moravec. To compare different machines, Moravec defined computing power as the amount of information delivered per second by the machine.<sup>15</sup>

This can then be put on a standardized basis by considering words with a standard length of 32 bits (equivalent to a nine-digit integer), and instructions with a length of one word. Moravec assumed that there were 32 instructions, and included measures on addition and multiplication time, which were weighted seven to one in the operation mix. Using this definition, the information-theoretic definition of performance is

$$\text{Computer power (Moravec)} = \frac{\{[6 + \log_2 (\text{memory}) + \text{word length}/2]\}}{[(7 \times \text{add time} + \text{mult time})/8]}$$

The attractiveness of this approach is that each of these parameters is available for virtually all computers back to 1940, and can be estimated or inferred for manual calculations, abacuses, and many early calculators. The disadvantages are that it omits many of the important operations of modern computers, it considers only machine-level operations, and it cannot incorporate the advantages of modern software, higher-level languages, and operating systems.

## KNIGHT'S MEASURE

One of the earliest studies of computer performance was by Kenneth Knight of RAND in 1966.<sup>16</sup> He wanted to go beyond the simplest measures of addition and multiplication time and did a number of ex-

<sup>15</sup> See Moravec, *Mind Children*, especially appendix A2 and p. 63f.

<sup>16</sup> Knight, "Changes," *Datamation*, pp. 40–54, and "Evolving Computer Performance," pp. 31–35.

periments on the capacity of different machines to perform different applications. His formula for computer power was as follows

$$\text{Knight's Index of Computer Power} \approx 10^6 \{[(\text{word length} - 7)(\text{memory})] / [\text{calculation time} + \text{input-output time}]\}$$

Knight's formula is quite similar to Moravec's except that he includes a larger number of variables and particularly because he calibrates the parameters to the actual performance of different machines.

#### MIPS

One of the earliest benchmarks used was MIPS, or millions of instructions per second. In simple terms, instructions per second measures the number of machine instructions that a computer can execute in one second. This measure was developed to compare the performance of mainframe computers. The most careful studies used weighted instruction mixes, where the weights were drawn from the records of computer centers on the frequency of different instructions. These benchmarks were probably the only time something approaching the ideal measure described previously was constructed.

A simplified description of MIPS is the following. For a single instruction

$$\text{MIPS} = \text{clock rate} / (\text{cycles per instruction} \times 10^6)$$

To understand the logic of this measure, recall that computers that use the von Neumann architecture contain an internal clock that regulates the rate at which instructions are executed and synchronizes all the various computer components. The speed at which the microprocessor executes instructions is its "clock speed." For most personal computers up to around 2000, operations were performed sequentially, once per clock tick.<sup>17</sup> An instruction is an order given to a computer processor by a computer program. Computers with complex instruction sets might have between 200 and 400 machine-language instructions, whereas computers with reduced instruction sets would have only 30 to 50 unique instructions.

Instructions differ in terms of the size of the "word" that is addressed. In the earliest computers (such as the Whirlwind I), words were as short as 16 binary digits or five decimal digits. Most personal

<sup>17</sup> Many of the major topics in computer architecture can be found in books on computer science. For example, see Schneider and Gersting, *Invitation*.

TABLE 4  
SUITE OF PROGRAMS USED FOR SPEC2000 BENCHMARK

SPECint2000	SPECfp2000
Compression	Physics: Quantum chromodynamics
FPGA circuit placement and routing	Shallow water modeling
C programming language compiler	Multigrid solver: 3D potential field
Combinatorial optimization	Partial differential equations
Game playing: Chess	3D graphics library
Word processing	Computational fluid dynamics
Computer visualization	Image recognition/neural networks
Perl programming language	Seismic wave propagation simulation
Group theory, interpreter	Image processing: Face recognition
Object-oriented database	Computational chemistry
Compression	Number theory/primality testing
Place and route simulator	Finite-element crash simulation
	Nuclear physics accelerator design
	Meteorology: Pollutant distribution

*Note:* This table shows the benchmarks used to evaluate different computers according to the SPEC2000 benchmark. The first set use largely integer applications while the second are largely floating-point scientific applications.

*Source:* Henning, "SPEC CPU2000," p. 29.

computers today use 32-bit words, while mainframes generally employ 64-bit words.

#### MODERN BENCHMARK TESTS

Measures such as additions or instructions per second or more complete indexes such as those of Knight or Moravec clearly cannot capture today's complex computational environment. Computers today do much more than bookkeeping, and a performance benchmark must reflect today's mix of activities rather than that of a century ago. For this purpose, we turn to modern benchmark tests.

A *benchmark test* is an index that measures the performance of a system or subsystem on a well-defined set of tasks. Widely used benchmarks for personal computers today are those designed by SPEC, or the Standard Performance Evaluation Corporation. As of mid-2006, the version used for personal computers was SPEC CPU2000.<sup>18</sup> SPEC CPU2000 is made up of two components that focus on different types of compute intensive performance: SPECint2000 for measuring and comparing computer-intensive integer computation and SPECfp2000 for measuring computer-intensive floating-point computation.

Table 4 shows the suite of activities that SPEC2000 tests. These are obviously not routine chores. The benchmark fails to follow the elemen-

<sup>18</sup> See <http://www.spec.org/osg/cpu2000/>.

tary rule of ideal indexes in that the performance on different benchmarks is clearly not weighted by the economic importance of different applications. I discuss below the relationship between the SPEC and other benchmarks. To make current tests comparable with early ones, ratings have been set by comparing the rating of a machine with the rating of a benchmark machine.

#### MEASURE OF COMPUTER PERFORMANCE

This study is an attempt to link together computational performance of different machines from the nineteenth century to the present. A unit of computer performance is indexed so that manual computations are equal to one. A standard modern convention is that the VAX 11-780 is designated as a one MIPS machine. In our units, the VAX 11-780 is approximately 150 million times as powerful as manual computations. Different modern benchmarks yield different numbers, but they are essentially scalar multiples of one another.

Constructing metrics of performance is difficult both because the tasks and machines differ enormously over this period and because measures of performance are very sketchy before 1945. The data since 1945 have been the subject of many studies since that period. Data for this study for computers from 1945 to 1961 were largely drawn from technical manuals of the Army Research Laboratory, which contain an exhaustive study of the performance characteristics of systems from ENIAC through IBM-702.<sup>19</sup> Additionally, studies of Kenneth Knight provided estimates of computer power for the period 1945 through 1966. Data on the performance of computers through 2003 have been carefully compiled by John C. McCallum and are available on the web.<sup>20</sup> Machines since 2003 were evaluated by the author. All data on performance and cost are available in the background document.<sup>21</sup>

Reliable data for the earliest calculators and computers (for the period before 1945) were not available in published studies. With the help of Eric Weese of Yale University, data from historical sources on the performance of 32 technologies from before 1940 were obtained, for which 12 have performance and price data that I consider reasonably reliable. I will discuss the data on the early technologies because these are the major original data for the present study.

<sup>19</sup> See particularly Weik, *Survey*. This was updated in Weik, *Third Survey*.

<sup>20</sup> See <http://www.jcmitt.com/cpu-performance.htm>; as well as Nordhaus, Online Appendix, page "MacCallum."

<sup>21</sup> Nordhaus, Online Appendix, page "Abacus."

The data on manual calculations were taken from a Burroughs monograph, from estimates of Moravec, and from tests by the author.<sup>22</sup> The computational capabilities of the abacus are not easily measured because of the paucity of users in most countries today. One charming story reports a Tokyo competition in 1946 between the U.S. Army's most expert operator of the electric calculator in Japan and a champion operator of the abacus in the Japanese Ministry of Postal Administration. According to the report, the addition contest consisted of adding 50 numbers each containing three to six digits. In terms of total digits added, this is approximately the same as the tests shown in Figure 1. The abacus champion completed the addition tasks in an average of 75 seconds, while the calculator champion required 120 seconds. They battled to a standoff in multiplication and division. The abacus expert won four of the five contests and was declared the victor.<sup>23</sup>

This comparison suggests that, in the hands of a champion, the abacus had a computer power approximately four and a half times that of manual calculation. Given the complexity of using an abacus, however, it is unlikely that this large an advantage would be found among average users. We have reviewed requirements for Japanese licensing examinations for different grades of abacus users from the 1950s. These estimates suggest that the lowest license level (third grade) has a speed approximately 10 percent faster than manual computations.<sup>24</sup>

We have estimated the capabilities of early machines based on then-current procedures. For example, many of the early machines were unable to multiply. We therefore assume that multiplication was achieved by repeated addition. Additionally, the meaning of memory size in early machines is not obvious. For machines that operate by incrementation, we assume that the memory is one word. There are major discrepancies between different estimates of the performance of early machines, with estimates varying by as much as a factor of three. Given the difficulties of collecting data on the earliest machines, along with the problems of making the measures compatible, we regard the estimates for the period before 1945 as subject to large errors.

The only other important assumptions involve constructing the cost per operation. These calculations include primarily the cost of capital. The data on prices and wage rates were prepared by the author and are from standard sources, particularly the U.S. Bureau of Labor Statistics and the U.S. Bureau of Economic Analysis. We have also included estimates of operating costs as these appear to have been a substantial

<sup>22</sup> Nordhaus, Online Appendix, page "Manual."

<sup>23</sup> The contest and its results are described in Kojima, *Japanese Abacus*.

<sup>24</sup> Kojima, *Japanese Abacus*.

TABLE 5  
OVERALL IMPROVEMENT IN COMPUTER POWER FROM MANUAL CALCULATIONS  
UP TO THE PRESENT

Improvement from Manual (1850) to 2006	Improvement	
	Ratio	Annual Growth Rate (percent per year)
Computer power (MCPS per second)	2,050,000,000,000	18.3
Price per calculation (MCPS per 2003\$)	7,100,000,000,000	19.1
Labor cost of computation (MCPS per hour)	73,000,000,000,000	20.6

Source: Nordhaus, *Online Appendix*, page “Data.”

fraction of costs for many computers and may be important for recent computers. For the capital cost, we use the standard user cost of capital formula with a constant real interest rate of 10 percent per year, an exponential depreciation rate of 10 percent per year, a utilization factor of 2,000 hours per year, and no adjustment for taxes. These assumptions are likely to be oversimplified for some technologies, but given the pace of improvement in performance, even errors of 10 or 20 percent for particular technologies will have little effect on the overall results.

## RESULTS

### *Overall Trends*

I now discuss the major results of the study. Table 5 shows a summary of the overall improvement in computing relative to manual calculations and the growth rates in performance. The quantitative measures are computer power, the cost per unit computer power in terms of the overall price level, and the cost of computation in terms of the price of labor. The overall improvements relative to manual computing range between two and 73 trillion depending upon the measure used. For the period 1850 (which I take as the birth of modern computing) to 2006, the compound logarithmic growth rate is around 20 percent per year.

We now discuss the results in detail. Start with Figure 2, which shows the results in terms of pure performance—computing power in terms of computations per second. Recall that the index is normalized so that manual computation is one. Before World War II, the computation speeds of the best machines were between ten and 100 times the speed of manual calculations. There was improvement, but it was relatively slow. Figure 3 shows the trend in the cost of computing over the last century and a half. The prices of computation begin at around \$500 per MCPS for manual computations and decline to around  $\$6 \times 10^{-11}$

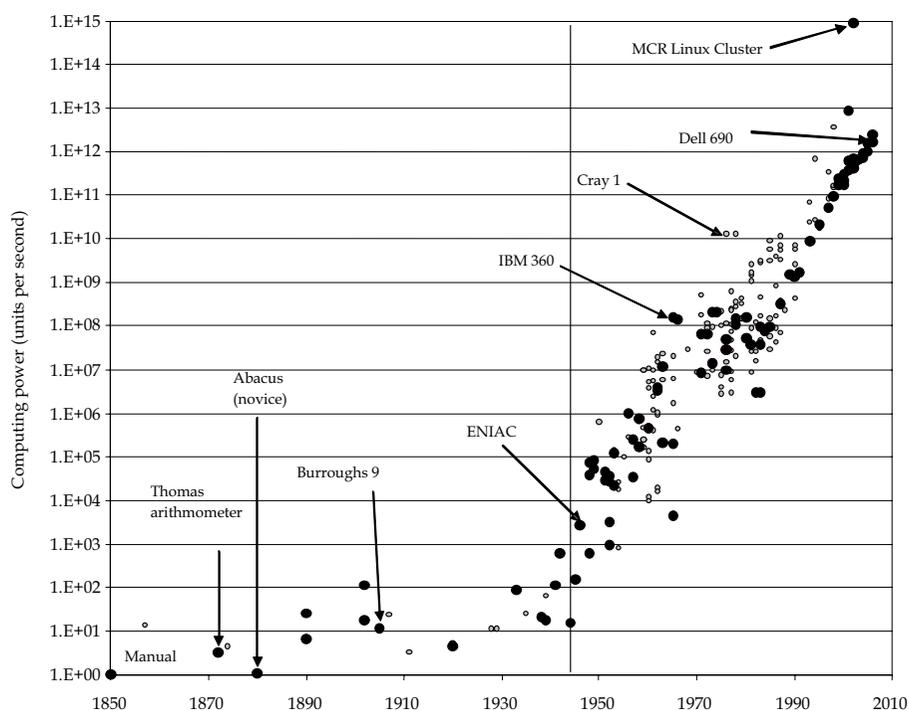


FIGURE 2  
THE PROGRESS OF COMPUTING POWER MEASURED IN COMPUTATIONS PER SECOND (CPS)

*Notes:* The measure shown here is the index of computing power. For a discussion of the definition, see the text. The series is defined so that manual computing has a value of one. The large circles are estimates that have been judged relatively reliable, and the small circles are estimates in the literature that have not been independently verified. The vertical line is placed at 1944, which is the estimated breakpoint in productivity growth.

*Source:* As described in the text.

per MCPS by 2006 (all in 2006 prices), which is a decline of a factor of seven trillion.

Table 6 shows five different measures of computational performance, starting with manual computations through 2006. The five measures are computer power, cost per unit calculation, labor cost per unit calculation, cycles per second, and rapid memory. The general trends are similar, but different measures can differ substantially. One important index is the relative cost of computation to labor cost. This is the inverse of total labor productivity in computation, and the units are therefore CPS per hour of work.<sup>25</sup> Relative to the price of labor,

<sup>25</sup> The advantages of using the wage rate as a deflator are twofold. First, it provides a measure of the relative price of two important inputs (that is, the relative costs of labor and computation).

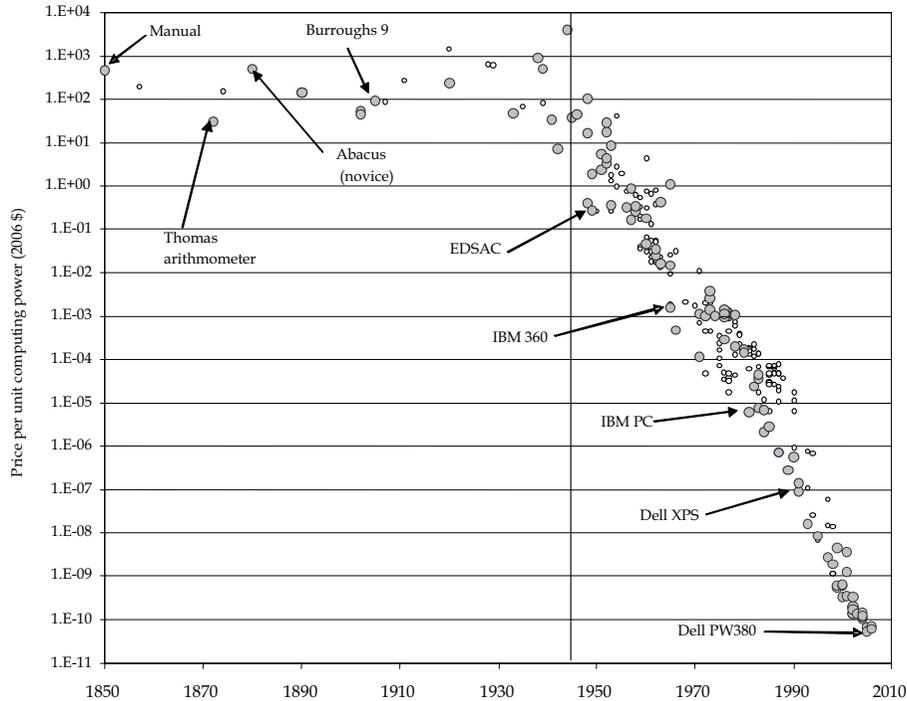


FIGURE 3

THE PROGRESS OF COMPUTING MEASURED IN COST PER COMPUTATION PER SECOND DEFLATED BY THE PRICE INDEX FOR GDP IN 2006 PRICES

*Notes:* The larger circles are estimates that have been judged relatively reliable, while the small circles are estimates in the literature that have not been independently verified. The vertical line is placed at 1944, which is the estimated breakpoint in productivity growth.

*Source:* As described in the text.

computation has become cheaper by a factor of  $7.3 \times 10^{13}$  compared to manual calculations. Given the enormous decrease in computational cost relative to labor cost, it can hardly be surprising that there has been a rapid increase in the computer-intensity of production (as measured by the ratio of computer capital to output or computer capital to labor), a point I return to in the final section.

### *Trends for Different Periods*

We next examine the progress of computing for different subperiods. The major surprise, clearly shown in Figures 2 and 3, is the discontinuity

---

Additionally, the convention of using a price index as a deflator is defective because the numerator is also partially contained in the denominator.

TABLE 6  
BASIC PERFORMANCE CHARACTERISTICS BY EPOCHS OF COMPUTING

Period	Computer Power (units per second)	Total Cost per Million Unit Computer Power (2006 \$)	Labor Cost of Computation (hours per unit computer power)	Cycle Speed (cycles per second)	Rapid Access Memory (bits)
Manual	1.00E+00	4.58E+02	2.78E+02	1.00E-01	3.32E+00
Late 19th C.	6.48E+00	1.46E+02	5.34E+01	2.00E-01	2.00E+01
1900–1939	1.78E+01	9.16E+01	3.11E+01	7.50E-01	2.66E+02
1940–1949	1.67E+03	2.57E+01	3.17E+00	2.50E+04	3.71E+03
1950–1959	1.18E+05	7.64E+01	7.09E-02	1.80E+05	3.73E+04
1960–1969	2.92E+06	3.38E-02	2.71E-03	1.00E+06	1.02E+06
1970–1979	7.48E+07	5.92E-04	4.00E-05	5.56E+06	4.10E+06
1980–1989	1.50E+08	4.73E-05	3.27E-05	8.02E+06	2.05E+06
1990–1999	4.02E+10	2.08E-08	1.44E-09	1.82E+08	5.12E+08
2000–2006	8.39E+11	1.37E-10	8.26E-12	1.80E+09	1.64E+10

Note: Each year takes the median of computers or devices for that period.

that took place around World War II. Table 6 shows data on performance of machines in different periods, while Table 7 shows the logarithmic annual growth rates between periods (defining manual calculations as the first period). Table 7 indicates modest growth in performance from manual computation until the 1940s. The average increase in computer productivity shown in the first three columns of the first row of Table 7—showing gains of around 3 percent per year—was probably close to the average for the economy as a whole during this period.

Statistical estimates of the decadal improvements are constructed using a log-linear spline regression analysis. Table 8 shows a regression of the logarithm of the constant-dollar price of computer power with decadal trend variables. The coefficient is the logarithmic growth rate, so to get the growth rate for a period we can sum the coefficients up to that period. The last column of Table 8 shows the annual rates of improvement of computer performance. All measures of growth rates are logarithmic growth rates.<sup>26</sup>

<sup>26</sup> The growth rates are instantaneous or logarithmic growth rates, which are equivalent to the derivatives of the logarithms of series with respect to time for smooth variables. This convention is used to avoid the numerical problems that arise for high growth rates.

A word is in order for those not accustomed to logarithmic growth rates: These will be close to the conventional arithmetic growth rate for small numbers (2 or 3 percent per year) but will diverge significantly for high growth rates. For example, an arithmetic growth rate of 100 percent per year is equivalent to a logarithmic rate of 0.693. That is,  $e^{0.693} = 2$ . A further warning should be given on negative growth rates. There is no difficulty in converting negative to positive rates as long as logarithmic growth rates are used. However, in using arithmetic growth rates, decline rates may look significantly smaller than the corresponding growth rate. For example, a logarithmic growth rate of  $-0.693$  represents a decline rate of 50 percent per year; that is,  $e^{-0.693} = 0.5$ .

TABLE 7  
GROWTH RATES OF DIFFERENT PERFORMANCE CHARACTERISTICS OF  
PERFORMANCE IN DIFFERENT EPOCHS OF COMPUTING  
(average annual logarithmic growth rates, percent)

Period	Computer Power (units per second)	Total Cost per Million Unit Computer Power (2006 \$)	Labor Cost of Computation (hours per unit computer power)	Cycle Speed (cycles per second)	Rapid Access Memory (bits)
From Manual to late 19th C	6.2	-3.8	-5.5	2.3	6.0
From late 19th C to 1900-39	2.4	-1.1	-1.3	3.1	6.1
1900-39 to 1940-49	19.0	-5.3	-9.6	43.7	11.1
1940-49 to 1950-59	47.2	-38.9	-42.1	21.8	25.6
1950-59 to 1960-69	44.0	-42.7	-44.7	23.5	45.4
1960-69 to 1970-79	24.9	-31.1	-32.4	13.2	10.7
1970-79 to 1980-89	7.9	-28.8	-28.5	4.2	-7.9
1980-89 to 1980-99	53.7	-74.2	-74.2	30.0	53.0
1990-99 to 2000-06	36.7	-60.6	-62.3	27.7	41.8

Source: Table 6.

The regression analysis shows that the explosion in computer power, performance, and productivity growth began in the mid-1940s. A Chow test for stability of coefficients find the maximum-likelihood year for the break in trend was 1944, but the data cannot distinguish that year from neighboring years with a high degree of statistical significance. Tables 7 and 8 provide slightly different estimates of the subperiod growth rates, but it is clear that productivity growth was extremely rapid during virtually the entire period since 1945. Using decadal trend-break variables, as shown in Table 8, we find highly significant positive coefficients for the dummy variables beginning in 1945 and in 1985 (both indicating acceleration of progress). The only period when progress was slow (only 22 percent per year!) was during the 1970s. Table 7 uses a different methodology for examining subperiods. It shows a slowing in the 1960-1969 and 1970-1979 periods. We were unable to resolve the timing and cause of the slowdown in the 1960-1979 subperiod, and this is left as an open question.

The rapid improvement in computer power is often linked with "Moore's Law." This derives from Gordon Moore, co-founder of Intel, who observed in 1965 that the number of transistors per square inch on integrated circuits had doubled every year since the integrated circuit was invented. Moore predicted that this trend would continue for the foreseeable future. When he revisited this question a decade later, he thought that the growth rate had slowed somewhat and forecast that

TABLE 8  
REGRESSION ANALYSIS FOR TRENDS IN COMPUTING POWER

Variable	Coefficient	Std. Error	<i>t</i> -Statistic	Decline Rate per Decade (percent per year)
<i>Year</i>	-0.006	0.010	-0.632	-0.6
<i>DUM1945</i>	-0.445	0.065	-6.798	-45.1
<i>DUM1955</i>	0.026	0.108	0.240	-42.5
<i>DUM1965</i>	0.062	0.098	0.629	-36.3
<i>DUM1975</i>	0.147	0.078	1.875	-21.6
<i>DUM1985</i>	-0.527	0.083	-6.333	-74.3
<i>DUM1995</i>	0.135	0.114	1.190	-60.8

*Notes:* This table shows the results of a spline regression of the logarithm of the deflated price of computer power as a function of year and time dummies. The time variables are zero up to the year and then increment by one per year after the date. The last column shows the cumulative sum, which can be interpreted as the rate of decline in cost for the decade or period shown in the first column.

*Year* is calendar year. *DUM*"year" is a variable that takes on a value of zero up to "year," and value of year minus "year" after "year."

The growth rate is logarithmic.

The dependent variable is ln(cost per CPS in 2006 dollars).

*N* equals 235.

doubling every 18 months was a likely rate for the future (46 percent logarithmic growth). Two remarks arise here. First, it is clear that rapid improvements in computational speed and cost predated Moore's forecast. For the period 1945–1980, cost per computation declined at 37 percent per year, as compared to 64 percent per year after 1980. Second, computational power actually grows more rapidly than Moore's Law would predict because computer performance is not identical to chip density. From 1982 to 2001, the rate of performance as measured by computer power grew 18 percent per year faster than Moore's Law would indicate.

One of the concerns with the approach taken in this study is that our measures might be poor indexes of performance. We have compared MCPS with both addition time and cycle time (the latter comparison is shown in Tables 6 and 7). Both simple proxies show a very high correlation with our synthetic measure of MCPS over the entire period. Computer power grows at very close to the speed of addition time (for observations from 1900 to 1978) but 10 percent per year more rapidly than cycle speed (1938–2006).

In this regard, it is natural to ask whether the changing character of computers is likely to bias the estimates of the price of computer power. The earliest calculators had very low capability relative to modern computers, being limited to addition and multiplication. Modern computers perform a vast array of activities that were unimaginable a century ago

(see Table 4). In terms of the ideal measure described above, it is likely that standard measures of performance are biased *downward*. If we take an early output mix—addition only—then the price index changes very little, as discussed in the last paragraph. On the other hand, today's output bundle was infeasible a century ago, so a price index using today's bundle of output would have fallen even faster than the index reported here. Put differently, a particular benchmark only includes what is feasible, that is, tasks that can be performed in a straightforward way by that year's computers and operating systems. Quantum chromodynamics is included in SPEC 2000, but it would not have been dreamt of by Kenneth Knight in his 1966 study. This changing bundle of tasks suggests that, if anything, the price of computation has fallen even faster than the figures reported here.

#### ALTERNATIVE APPROACHES

##### *Comparison of Alternative Modern Benchmark Tests*

Using direct measures of computer performance raises two major problems. First, a properly constructed benchmark should weight the different tasks according to their economic importance, but this property is satisfied by none of the current benchmark tests. For example, as shown in Table 4, the SPEC benchmark that is widely used to test PCs contains several exotic tasks, such as quantum chromodynamics, which are probably not part of the family computer hour. Most benchmarks simply apply equal geometric weights to the different tasks. Second, the rapid evolution of computer performance leads to rapid changes in the tasks that the benchmarks actually measure. For example, the SPEC performance benchmark has been revised every two or three years. In one sense, such changes represent a kind of chain index in tasks; however, because tasks are not appropriately weighted, it is impossible to know whether the chaining improves the accuracy of the indexes.

Eric Weese and I investigated the results of using different benchmark tests over the last decade. For this purpose, we examined the SPEC benchmarks; a series of tests known as WorldBench, which have been published by *PC World*; and SYSmark98, a measure that evaluates performance for 14 applications-based tasks. To illustrate how PC benchmarks work, the SYSmark98 test of office productivity is the harmonic mean of the time to open and perform set tasks on the following programs: CorelDRAW 8, Microsoft Excel 97, Dragon Systems NaturallySpeaking 2.02, Netscape Communicator 4.05 Standard Edi-

tion, Caere OmniPage Pro 8.0, Corel Paradox 8, Microsoft PowerPoint 97, and Microsoft Word 97.

We first examined 30 machines for which both PC benchmarks had results over the period December 1998 to November 1999. The two benchmarks were reasonably consistent, with a correlation of 0.962 in the logarithm of the benchmark scores over the 30 machines. However, as shown in comparison one of Table 9, the rate of improvement of the two indexes differed markedly, with the SYSmark98 showing a 38 percent per year improvement over the 11 month period, while the WorldBench tests showed a 24 percent per year improvement for the same machines. This difference was found even in the individual benchmarks (e.g., the results for Excel 97), and queries to the sources produced no reasons for the discrepancies.<sup>27</sup>

A second comparison was between the SPEC benchmark results and the WorldBench tests. For this comparison, we were able to find seven machines that were tested for both benchmarks over a period of two years, using the 1995 SPEC test and three different WorldBench tests. For these machines, as shown in comparison two of Table 9, the rate of improvement of the SPEC and WorldBench tests were virtually identical at 67 and 66 percent per year, respectively.

The final test involved a comparison of the WorldBench score with the improvement in computations per second calculated for this article. For this purpose, we gathered different WorldBench tests for the period from 1992 to 2002 and spliced them together to obtain a single index for this period. We then calculated the growth of WorldBench performance per constant dollar and compared this to the growth of CPS per constant dollar from the current study. As shown in comparison three, the WorldBench performance per real dollar over 1992–2002 showed a 52 percent per year increase. This compares with a 62 percent per year increase for the computers in our data set over the same period.

To summarize, we have investigated the results of alternative benchmarks tests. None of the benchmarks is well constructed because the weights on the different tasks do not reflect the relative economic importance of tasks. We found some discrepancies among the different benchmarks, even those that purport to measure the same tasks. The WorldBench test, which is oriented toward PCs, showed slower improvements in constant-dollar performance over the 1992–2002 period than the CPS measure constructed for this study. However, an alternative test, SYSmark, show more rapid growth than the WorldBench and

<sup>27</sup> For example, we compared the raw scores for the two benchmarks for six identical machines and three identical programs. The harmonic means differed by as much as 17 percent between the two sets of tests.

TABLE 9  
COMPARISON OF DIFFERENT BENCHMARKS

	Comparison 1 Dec. 1998–Nov. 1999		Comparison 2 1995–1996		Comparison 3 1992–2002	
	WorldBench per Constant Dollar	YSMark98 per Constant Dollar	WorldBench per Constant Dollar	SPEC per Con- stant Dollar	WorldBench per Constant Dollar	Computations per Second Constant Dollar
Number of computers	30	30	7	7	47	37
Logarithmic growth rate						
Coefficient	0.24	0.38	0.66	0.67	0.52	0.62
Standard error	0.19	0.22	0.05	0.2	0.02	0.03
Arithmetic growth rate in computer performance per year (%)	27.1	46.2	93.5	95.4	68.2	85.9

*Notes:* This table shows the results of three sets of comparison between alternative benchmark tests of computer performance. In each case, the variable examined is computer performance per constant dollar, where the deflator is the consumer price index. The first two comparisons use exactly the same machines, while comparison 3 uses different machines over the same period. The different benchmarks are described in the text.

was more consistent with the CPS measure in this article and with the SPEC benchmark. In any case, the improvement in constant dollar performance was extremely rapid, with the lower number being a 52 percent per year logarithmic increase for the WorldBench and the higher number being a 62 percent per year increase over the last decade for the CPS.

#### *Comparison with a Direct Measure of Performance*

The long-term performance measures used in this study are constructed by chaining together several different benchmark tests. Because they use different weights and underlying benchmarks, we might be concerned that the actual performance diverges substantially over time from our chained index, particularly when the changes are by a factor of a trillion or more.

To examine the potential long-term bias, I undertook a simple set of multiplication and addition experiments with present-day personal computers (the actual calculations were performed by Ray Fair). Using a small Fortran program, we performed 100 billion multiplications of three three-digit numbers and 1 trillion additions of two five-digit numbers. These calculations took 60 and 315 seconds respectively on a 1.7 GHz Dell machine. Putting these numbers into the Moravec formula gives a computer power for this machine that is  $4.7 \cdot 10^{11}$  larger than manual computations. The same machine is rated as  $4.1 \cdot 10^{11}$  times more powerful than manual computations in our index of computer power. The estimated computer power of the chained index is therefore extremely close to the actual performance for addition and multiplication. This calculation suggests that the constructed index is reasonably consistent with actual computer performance and that no obvious drift has arisen from chaining the different measures together.

#### *Comparison with Alternative Indexes of Computer Prices*

Economists today tend to favor the use of hedonic or constant-quality price indexes to measure improvements. The hedonic approaches measure the change in the “quantity” of goods by examining the change in characteristics along with measures of the importance of the different characteristics.<sup>28</sup>

A pioneering study that investigated hedonic prices of performance was undertaken by Paul Chwelos. He investigated the characteristics of

<sup>28</sup> There are many excellent surveys of hedonic methods. A recent National Academy of Sciences report has a clear explanation of different approaches. See Schultze and Mackie, *At What Price*.

computers that were important for users and information scientists in 1999 and found the top six characteristics were performance, compatibility, RAM, network connectivity, industrial standard components, and operating system.<sup>29</sup> He then estimated the change in the cost of providing the bundle.

Clearly, such an approach is not feasible over the long span used here. In the present study, we examined only the price of a single characteristic, performance. This decision reflects the fact that only two of the six performance characteristics discussed in the last paragraph (performance and RAM) can be tracked back for more than a few decades. Network connectivity is a brand-new feature, while operating systems have evolved from tangles of wires to Windows-type operating systems with tens of millions of lines of high-level code. This discussion indicates that computers have experienced not only rapid improvements in speed but also in breadth through a growing array of goods and services.

How do the performance-based indexes used here compare with price indexes for computers? A summary table of different price indexes for recent periods is provided in Table 10. There are six variants of computer price indexes prepared by the government, either for the national income and product accounts by the Bureau of Economic Analysis (BEA) or for the Producer Price Index by the Bureau of Labor Statistics.<sup>30</sup>

During the 1969–2004 period, for which we have detailed price indexes from the BEA, the price index for computers fell by 23 percent per year relative to the GDP price index (using the logarithmic growth rate), while the real BLS price index for personal workstations and computers fell by 31 percent. Academic studies, using hedonic approaches or performance measures, show larger decreases, between 35 and 40 percent. Our real price index of the price of computer power fell by between 50 and 58 percent depending upon the subperiod.

How might we reconcile the significant discrepancy between the hedonic measures and the performance-based prices reported here? A first possible discrepancy arises because government price indexes for computers are based on the prices of *inputs into computers*, while the measures presented here are indexes of the *cost of specified tasks*. The hedonic measures will only be accurate to the extent that the prices of

<sup>29</sup> See Chwelos, *Hedonic Approaches*, p. 43. Performance was defined as a “characteristic of the a number of components: CPU (generation, Level 1 cache, and clock speed), motherboard architecture (PCI versus ISA) and bus speed, quantity and type of Level 2 cache and RAM, type of drive interface (EIDE versus SCSI).”

<sup>30</sup> See Landefeld and Grimm, “Note,” pp. 17–22, for a discussion and a compilation of studies. The data are available at <http://www.bea.doc.gov/>.

TABLE 10  
COMPARISON OF PRICE INDEXES FOR DIFFERENT STUDIES

Study	Period	Method	Rate of Real Price Decline (percent per year) <sup>g</sup>	Source
Government price data				
Price index for computers and peripherals (BEA)	1990–2004	Hedonic	17.8	a
	1969–2004	Hedonic	18.7	a
Price index for personal computers (BEA)	1990–2004	Hedonic	28.6	b
Producer price index (BLS)				
Semiconductors and related devices	1990–2004	Hedonic	50.5	c
Personal workstations and computers	1993–2004	Hedonic	31.2	c
Academic studies				
Berndt and Rappaport, personal computers	1989–1999	Hedonic	38.3	d
Chwelos, desktop computers	1990–1998	Performance	37.2	e
This study				
Price of computer power (\$ per MCPS)	1969–2005	Performance	50.7	f
Price of computer power (\$ per MCPS)	1990–2002	Performance	57.5	f

*Note:* This table shows estimates of the decline in prices of computers from different studies and methodologies.

*Sources:*

<sup>a</sup> BEA web page at [www.bea.gov](http://www.bea.gov), table 5.3.4.

<sup>b</sup> BEA web page at [http://www.bea.gov/bea/dn/info\\_comm\\_tech.htm](http://www.bea.gov/bea/dn/info_comm_tech.htm).

<sup>c</sup> BLS web page at [www.bls.gov](http://www.bls.gov).

<sup>d</sup> Landefeld and Grimm, “Note.”

<sup>e</sup> Chwelos, *Hedonic Approaches*.

<sup>f</sup> From regression of logarithm of price on year for period.

<sup>g</sup> All prices use price index for GDP to deflate nominal prices.

components accurately reflect the marginal contribution of different components to users’ valuation of computer power. It is worth noting that current government hedonic indexes of computers contain *no* performance measure.<sup>31</sup>

<sup>31</sup> The variables in an earlier BLS hedonic regression for personal desktop computers (designed in 1999 but discontinued after 2003) contained one performance proxy (clock speed), two performance-related proxies (RAM and size of hard drive), an array of feature dummy variables (presence of Celeron CPU, ZIP drive, DVD, fax modem, speakers, and software), three company dummy variables, and a few other items. It contained no performance measures such as the SPEC benchmark. The new BLS pricing approach contains no performance measures at all and instead uses attribute values available on the Internet as a basis to determine appropriate quality adjustments amounts.

A second and more important difference is that computers increasingly are doing much more than computing, so the prices used in this study include many non-computational features. To illustrate, in late 2005, a Intel® Pentium® 4 Processor 630 with HT (3GHz, 2M, 800MHz FSB) was priced at \$218 while the Dell OptiPlex™ GX620 Mini-Tower personal computer in which it was embedded cost \$809. The \$591 difference reflects ancillary features such as hard-drives, ports, CD/DVD readers, pre-loaded software, assembly, box, and so forth. A perfectly constructed hedonic price index will capture this changing mix of components. To illustrate this point, assume that a 2005 computer is 25 percent computation and 75 percent ancillary enhancements while a 1965 computer such as the DEC PDP-8 was 100 percent computation. Excluding the ancillary items would change the real price decline from 45 percent per year to 48 percent over the four decades. It seems unlikely that the prices of the noncomputational components are falling as rapidly as the computational parts, a point that has been emphasized by Kenneth Flamm.<sup>32</sup> Hence, the discrepancy between current price measures and hedonic measures arises partly because “computers” are now doing much more than computing, and the price of the noncomputational components are falling less rapidly than the computational components.

### *Supercomputing*

While this study has emphasized familiar species of computers, it will be useful to devote a moment's attention to the elephants of the computer kingdom. Scientists and policy makers often emphasize supercomputing as the “frontier” aspect of computation, the “grand challenges of computation,” or the need for “high performance computing.” These are the romantic moon shots of the computer age. What are the grand challenges? Generally, supercomputers are necessary for the simulation or solution of extremely large nonlinear dynamic systems. In a recent report, the National Research Council listed some of the important current and prospective applications of supercomputers. These included defense and intelligence, climate prediction, plasma physics, transportation engineering, bioinformatics and computational biology, environmental science, earthquake modeling, petroleum exploration, astrophysics, nanotechnology, and macroeconomics.<sup>33</sup> To pick the second of these areas, climate prediction, there are enormous demands for improvements in modeling of climate systems and of the interactions be-

<sup>32</sup> Flamm, “Technological Advance,” pp. 13–61; and Flamm, “Digital Convergence,” p. 267.

<sup>33</sup> Graham, Snir, and Patterson, Eds., *Getting Up to Speed*, chapter 4.

tween oceans, the atmosphere, and the cryosphere; our understanding of many issues about the pace and impact of climate change will depend upon improving the models and the computers to solve the models.

The progress in supercomputing has paralleled that in smaller computers. As of November 2006, for example, the largest supercomputer (IBM's Blue Gene/L with 131,072 processors) operated at a maximum speed of 280,600 gigaflops (billions of floating-point operations per second or Gflops). Using a rough conversion ratio of 475 CPS per Flop, this machine is therefore approximately a 133,000,000,000 MCPS machine and therefore about 53,000 times more powerful than the top personal computer in our list as of 2006. The performance improvement for supercomputers has been tracked by an on-line consortium called "TOP500." It shows that the top machine's performance grew from 59.7 Gflops in June 1993 to 280,600 Gflops in November 2006.<sup>34</sup> Over this period, the peak performance grew at a rate of 87 percent per year. This is higher than the rate in our sample of smaller computers. However, it is likely that the performance was tuned to the benchmark, and the large systems are clearly not as versatile as personal computers.

The price of supercomputing is generally unfavorable relative to personal computers. IBM's stock model supercomputer, called "Blue Horizon," is clocked at 1,700 Gflops and had a list price in 2002 of \$50 million—about \$30,000 per Gflop—which makes it approximately 34 times as expensive on a pure performance basis as a Dell personal computer in 2004.

## CONCLUSIONS

The key purpose of this study is to extend estimates of the price of computers and computation back in time to the earliest computers and calculators as well as to manual calculations. Along the way, we have developed performance-based measures of price and output that can be compared with input-based or component-based measures. This final section discusses some reservations and then the major implications of the analysis.

Although we have provided performance-based measures of different devices, we note that the measures are generally extremely limited in their purview. They capture primarily computational capacity and generally omit other important aspects of modern computers such as connectivity, reliability, size, and portability. In one sense, we are comparing the transportation skills of the computer analogs of mice and men

<sup>34</sup> See [www.top500.org](http://www.top500.org).

without taking into account many of the “higher” functions that modern computers perform relative to mice like the IBM 1620 or ants like the Hollerith tabulator.

In addition, we emphasize that some of the data used in the analysis, particularly those for devices before 1945, are relatively crude. Additionally, the measures of performance or computer power used for early computers have been superceded by more sophisticated benchmarks. While conventional equivalence scales exist and are used when possible, the calibrations are imperfect. Subject to these reservations, six points emerge from the analysis.

First, there has been a phenomenal increase in computer power over the twentieth century. Performance in constant dollars has improved relative to manual calculations by a factor in the order of  $2 \times 10^{12}$  (2 trillion). Most of the increase has taken place since 1945, when the average rate of improvement has been 45 percent per year. The record shows virtually continuous extremely rapid productivity improvement over the last six decades. These increases in productivity are far larger than that for any other good or service in the historical record.<sup>35</sup>

Second, the data show a sharp break in trend around 1945—at the time when the technological transition occurred from mechanical calculators to what are recognizably the ancestors of modern computers. There was only modest progress—perhaps a factor of 10—in general computational capability from the skilled clerk to the mechanical calculators of the 1920s and 1930s. Around the beginning of World War II, all the major elements of the first part of the computer revolution were developed, including the concept of stored programs, the use of relays, vacuum tubes, and eventually the transistor, improved software, along with a host of other components. Dating from about 1945, computational speed increased and costs decreased rapidly up to the present. The most rapid pace of improvement was in the periods 1945–1955 and 1985–1995.

Third, these estimates of the growth in computer power, or the decline in calculation costs, are more rapid than price measures for computers used in the official government statistics. There are likely to be two reasons for the difference: first, the measures developed here are indexes of performance, whereas the approaches used by governments are based on the prices of components or inputs; and, second, “computers” to day are doing much more than computation.

<sup>35</sup> Scholars have sometimes compared productivity growth in computers with that in electricity. In fact, this is a cheetah-to-snails comparison. Over the half-century after the first introduction of electricity, its price fell 6 percent per year relative to wages, whereas for the six decades after World War II the price of computer power fell 47 percent per year relative to wages.

Fourth, the phenomenal increases of computer power and declines in the cost of computation over the last three decades have taken place through improvements of a given underlying technology: stored programs using the von Neumann architecture of 1946 and hardware based on Intel microprocessors descended from the Intel 4004 of 1971. The fact that this extraordinary growth in productivity took place in a relatively stable industry, in the world's most stable country, relying on a largely unchanged core architecture, is provocative for students of industrial organization to consider.<sup>36</sup>

Fifth, these results imply that there has been a rapid deepening of computer capital in the United States. Because of the growth in both the power and scope of computer power, the capital-labor ratio for computer capital has risen sharply. To provide an order-of-magnitude idea of the amount of capital deepening that has occurred, I estimate the amount of computer power available per hour of work. Using estimates of the number of machines and computer power per machine, I estimate that there was approximately 0.001 unit of (manual-equivalents of) computer power available per hour worked in 1900. That increased to about one unit of computer power per hour by the middle of the twentieth century. By 2005, computational power had increased to about  $10^{12}$  per hour worked.<sup>37</sup>

At the same time, and as a sixth point, this enormous growth in computer power does not imply that there were correspondingly large increases in economic welfare all along the way. The rapid increase in productivity reflected an equally rapid decline in the cost of computation, and the decline was probably matched by a similar decline in the marginal productivity of computing.

More importantly, the contribution of computer power to aggregate economic welfare depends upon the relative size of computer capital as well as its rate of productivity improvement. To a first approximation, the contribution of productivity growth of a particular kind of capital to overall economic welfare is proportional to the share of that particular capital in the total capital stock. In 1945, computer, software, and office equipment were only 0.1 percent of total capital stock, while by 2000 they were 2.3 percent of capital. Hence, even if productivity were growing at the same very rapid rate in the two periods, it would have contributed an order of magnitude less to economic welfare in the early period. While we have only fragmentary data on the value of calculating

<sup>36</sup> Bresnahan and Greenstein emphasize competition among platforms rather than among firms as key to the rapid productivity growth. See Bresnahan and Greenstein, "Technological Competition," pp. 1–40.

<sup>37</sup> Data on capital deepening is at Nordhaus, *Online Appendix*, pages "Capital\_deep."

machines around 1900, it appears that the share of calculating capital in 1900 was about one-twentieth of its share in 1945. These figures suggest why the rapid rates of computer performance became important for the overall economy only late in the twentieth century.<sup>38</sup>

What of the future? While forward-looking speculations might seem inappropriate in this JOURNAL, we would note that the history of computing to date shows no slackening of innovation in the fundamental computational processes or in applications of computation throughout the economy. Perhaps, aside from humans, computers and software are the ultimate general purpose technology.<sup>39</sup> They are a technology that has the potential for penetrating and fundamentally changing virtually every corner of economic life. At current rates of improvement, computers are approaching the complexity and computational capacity of the human brain.<sup>40</sup> Perhaps computers will prove to be the ultimate outsourcer.

<sup>38</sup> Data on aggregate capital and information capital are from BEA at <http://www.bea.gov/bea/dn/FA2004/SelectTable.asp>, table 2.1. Data for early calculators from Nordhaus, *Online Appendix*, pages “Quant\_History,” “Capital\_Deep,” and “Data.” The proposition in this paragraph assumes that capital goods and rental prices move inversely with productivity growth and that output is produced by a Cobb-Douglas production function.

<sup>39</sup> Bresnahan and Trajtenberg, “General Purpose Technologies,” pp. 83–108.

<sup>40</sup> See Moravec, *Mind Children*. For further details of the comparison, see Nordhaus, “Progress.”

## REFERENCES

- Bureau of Economic Analysis. <http://www.bea.gov/bea/dn/FA2004/SelectTable.asp>, table 2.1.
- Bresnahan, Timothy F., and Shane Greenstein. “Technological Competition and the Structure of the Computer Industry.” *The Journal of Industrial Economics* 47 (March 1999): 1–40.
- Bresnahan, Timothy F., and Manuel Trajtenberg. “General Purpose Technologies: ‘Engines of Growth’?” *Journal of Econometrics* 65, no. 1 (1995): 83–108.
- Burroughs Adding Machine Company. *A Better Day’s Work at a Less Cost of Time, Work and Worry to the Man at the Desk: In Three Parts Illustrated*, Third Edition. Detroit, MI, 1909.
- Chow, Gregory C. “Technological Change and the Demand for Computers.” *The American Economic Review* 57, no. 5 (December 1967): 1117–30.
- Chwelos, Paul. *Hedonic Approaches to Measuring Price and Quality Change in Personal Computer Systems*, Ph. D. Thesis, the University of Victoria, 1999.
- Cortada, James W. *Before the Computer*. Princeton, NJ: Princeton University Press, 1993.
- Flamm, Kenneth. “Technological Advance and Costs: Computers versus Communications.” In *Changing the Rules: Technological Change, International Competition, and Regulation in Communications*, edited by Robert W. Crandall and Kenneth Flamm, 13–61. Washington, DC: Brookings Institution, 1989.
- \_\_\_\_\_. “Digital Convergence? The Set-Top Box and the Network Computer.” In *Competition, Innovation and the Microsoft Monopoly: Antitrust in the Digital*

- Marketplace*, edited by J. A. Eisenach and T. M. Lenard, 267. Boston: Kluwer Academic Publishers, 1999.
- Freeth, T., Y. Bitsakis, X. Moussas, J. H. Seiradakis, A. Tselikas, H. Mangou, M. Zafeiropoulou, R. Hadland, D. Bate, A. Ramsey, M. Allen, A. Crawley, P. Hockley, T. Malzbender, D. Gelb, W. Ambrisco and M. G. Edmunds. "Decoding the Ancient Greek Astronomical Calculator Known as the Antikythera Mechanism." *Nature* 444 (30 November 2006): 587–91.
- Graham, Susan L., Marc Snir, and Cynthia A. Patterson, Eds. *Getting Up to Speed: The Future of Supercomputing*. Washington, DC: Committee on the Future of Supercomputing, National Research Council, National Academy Press, 2004.
- Henning, John L. "SPEC CPU2000: Measuring CPU Performance in the New Millennium." *Computer* 33, no. 7 (July 2000): 28–35.
- Johnston, Stephen. "Making the Arithmometer Count." *Bulletin of the Scientific Instrument Society* 52 (1997): 12–21, available online at <http://www.mhs.ox.ac.uk/staff/saj/arithmometer/>.
- Knight, Kenneth. "Changes in Computer Performance." *Datamation* 12, no. 9 (September 1966): 40–54.
- \_\_\_\_\_. "Evolving Computer Performance, 1963–1967." *Datamation* 14, no. 1 (January 1968): 31–35.
- Kojima, Takashi. *The Japanese Abacus: Its Use and Theory*. Rutland, VT: Charles E. Tuttle, 1954.
- Landefeld, J. Steven, and Bruce T. Grimm. "A Note on the Impact of Hedonics and Computers on Real GDP." *Survey of Current Business* 80 (December 2000): 17–22.
- Moravec, Hans. *Mind Children: The Future of Robot and Human Intelligence*. Cambridge, MA: Harvard University Press, 1988.
- Nordhaus, William. "The Progress of Computing." Cowles Foundation Discussion Paper No. 1324, September 2001.
- Schneider, G. Michael, and Judith L. Gersting. *An Invitation to Computer Science*. Pacific Grove, CA: Brooks/Cole, 2000.
- Schultze, Charles, and Christopher Mackie. *At What Price? Conceptualizing and Measuring Cost-of-Living and Price Indexes*. Washington, DC: National Research Council, 2002.
- Swade, Doron. *The Difference Engine*. New York: Viking Press, 2000.
- Triplett, Jack E. "Performance Measures for Computers." In *Deconstructing the Computer: Report of a Symposium*, edited by Dale W. Jorgenson and Charles W. Wessner, 97–140. Washington, DC: National Academy Press, 2005.
- Weik, Martin H. *A Survey of Domestic Electronic Digital Computing Systems, Ballistic Research Laboratories*, Report No. 971, December 1955, Department of the Army Project No. 5b0306002, Ordnance Research and Development Project No. Tb3-0007, Aberdeen Proving Ground, Maryland available at <http://ed-thelen.org/comp-hist/BRL.html>.
- \_\_\_\_\_. *A Third Survey of Domestic Electronic Digital Computing Systems*, Report No. 1115, March 1961, Ballistic Research Laboratories, Aberdeen Proving Ground, Maryland, available at <http://ed-thelen.org/comp-hist/BRL61.html#table-of-contents>.