Moore's Law

Moore's Law, proposed by Gordon E. Moore in 1965, is a theoretical rule stating that the number of transistors in an integrated circuit chips doubles about every 2 years (18 months to be exact), hence it will be an exponential growth. This law is generally accepted by many computer companies, since Moore's Law has a really profound impact on our current society, as it provides more powerful computers for relatively cheaper prices. This law can be observed in real life as computers nowadays have become smaller and smaller, but has an increasingly faster processing speed, as demonstrated in our smartphones[1], with relatively cheaper prices.

Rationale & Aim

My interest in Moore's Law developed when I first read Stephen Hawking's book "Brief Answers to the Big Questions", where the law is mentioned when stating the fact that the technology has rapidly developed at an unprecedented rate over the past few decades. This is also very relatable to my life. When I was born, there was hardly any touchscreen phones, but with merely few years later, there has been more touchscreen phones that are smaller and faster. As someone who has always been really interested in technology, this tremendous pace of growth has intrigued me massively.

Moore's Law is definitely an interesting concept and law that applies to computer growth. However, when talking about technological growth, the number of transistors in an integrated circuit chips may not be the only measure available. There are other measures such as computational operations or microprocessor clock speed that will also directly affect the performance or more accurately, processing speed, of a computer. Therefore, I am curious to see the extent to which Moore's Law can fit the growth of other measures like the ones suggested.

Therefore, in this exploration, I aim to develop a mathematical model and an equation describing the growth of computer's power in the form of microprocessor clock speed. Microprocessor clock speed is essentially the pulses per second generated by the processor. The higher the clock speed, the greater the computer power.[2] Additionally, I will produce an ideal model and equation, which shows the development of microprocessor clock speed if it strictly follows Moore's Law. This will allow

---

[1]"Moore's Law" (University of Missouri-St. Louis, May 17, 2013), http://www.umsl.edu/~siegelj/information_theory/projects/Bajramovic/www.umsl.edu/_abdcf/Cs4890/link1.html)

[2] Margaret Rouse, "What Is Clock Speed? - Definition from WhatIs.com," WhatIs.com, April 2005, https://whatis.techtarget.com/definition/clock-speed)

me to compare the two equations and models, in order to explore whether Moore's Law is accurate or not to describe the growth of computer power.

The microprocessor's clock speed data from 1976 to 2016 is attached below (table 1).

Year	Microprocessor clock speed (Hertz (pulses per second))
1976	1350000
1977	2060000
1978	2140000
1979	2290000
1980	1940000
1981	2410000
1982	2630000
1983	4070000
1984	5190000
1985	5890000
1986	7210000
1987	9430000
1988	12660000
1989	15630000
1990	19440000
1991	21180000
1992	29030000
1993	34150000
1994	53380000
1995	78040000
1996	140500000
1997	184280000
1998	337000000
1999	413680000
2001	1684000000
2002	2317000000
2003	3088000000
2004	3990000000
2005	5173000000

2006	5631000000
2007	6739000000
2010	11511000000
2013	19348000000
2016	28751000000

Table 1: Microprocessor Clock Speed (Data extracted from "Our World In Data"[3])

Part 1: Ideal Microprocessor's Clock Speed Model

According to Moore's Law, microprocessor clock speed should ideally double every 2 years. Thus, the sequence would be a geometric sequence as the next term of a sequence is found by multiplying the previous term by a common ratio.

The general formula for a term in a geometric sequence is given by:

$$u_n = u_1 r^{n-1}$$

In the formula:

- **u_n is the term number**
- **u_1 is the first term of the sequence**
  (This will be 1350000 as it is the first data value available, so it is assumed that there is no data before)
- **r is the common ratio**
  (This will be 2 as it is increased at a doubling rate.)
- **n is the -th term**

This will allow an ideal set of data from 1976 to 2016 to be calculated through the geometric formula. It would be the data if the microprocessor clock speed strictly doubles every 2 years. Substituting the available value into the formula:

$$u_n = 1350000 \times 2^{n-1}$$

For the purpose of this exploration, since it is stated that the growth will be doubling every 2 years. So, from 1976 to 2016, in a total of 40 years, there will be 20 two-years period. Thus, the highest n^{-th} term will be 21st, as the first term will be 1976, since it is the starting year. This equation will give the value for each term over the 40 years. For example, to find the value at the 13th term:

[3] Max Roser and Hannah Ritchie, "Technological Progress," Our World in Data, May 11, 2013, https://ourworldindata.org/technological-progress)

$$u_{13} = 1350000 \times 2^{13-1}$$

$$u_{13} = 1350000 \times 2^{12}$$

$$u_{13} = 552960000$$

Therefore, at year 2000 (the 13th term), the microprocessor clock speed should ideally be 552960000. Using the same method to find the rest of the data, table 2 shows the ideal set of data.

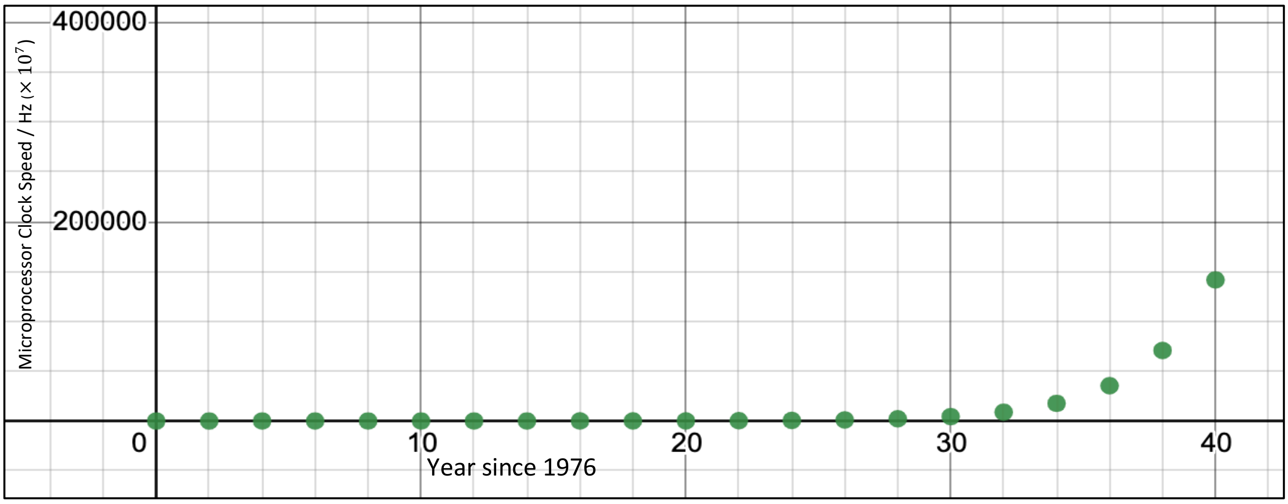
The microprocessor clock speed has been scaled down to $\times 10^7$ for clearer representation of data as the scale would be too big due to the y-value (microprocessor clock speed) reaching over 1 trillion.

n-th term	Year	Years since 1976	Microprocessor Clock Speed (Hz) $\times 10^7$
1st	1976	0	0.13500000
2nd	1978	2	0.27000000
3rd	1980	4	0.54000000
4th	1982	6	1.0800000
5th	1984	8	2.1600000
6th	1986	10	4.3200000
7th	1988	12	8.6400000
8th	1990	14	17.280000
9th	1992	16	34.560000
10th	1994	18	69.120000
11th	1996	20	138.24000
12th	1998	22	276.48000
13th	2000	24	552.96000
14th	2002	26	1105.9200
15th	2004	28	2211.8400
16th	2006	30	4423.6800
17th	2008	32	8847.3600
18th	2010	34	17694.720
19th	2012	36	35389.440
20th	2014	38	70778.880
21st	2016	40	141557.76

Table 2: Ideal Data Set

So, ideally by 2016, the microprocessor clock speed should be at around 141557.76×10^7 Hz if the increase in microprocessor clock speed follows Moore's Law strictly.

Plotting the model of the ideal data on a graph:



From the model alone, the limits of the graph of $f(x)$ can be easily observed:

$f(x)$ is defined as Microprocessor Clock Speed / Hz ($\times$ 10^7)

As $x \to +\infty$, $f(x) \to +\infty$

$\therefore \lim_{n \to -\infty} f(x)$ does not exist.

As $x \to -\infty$, $f(x) \to 0$

$\therefore \lim_{n \to \infty} f(x) = 0$, and the horizontal asymptote is y = 0.

The graph would be an exponential graph, due to its following characteristics:

- Continuous, smooth increasing trend
- The range is $y > 0$
- The domains are all real numbers
- The graph has a y-asymptote as x approaches 0.

Therefore, it could be represented with the general function for natural exponential:

$$f(x) = a \times e^{bx-c} + d$$

Before moving further, it is important to note that the model starts at around 0 in the y-axis, so it has not been transformed vertically, thus the constant d that controls this transformation will be 0. Also, there is an assumption that the function is not transformed horizontally, so the constant c which determine the horizontal transformation will be 0, thus there is less constant that may affect the shape of the graph in order to simplify the process. Thus, the function is now:

$$f(x) = a \times e^{bx}$$

- a which multiplies to vertically stretch up or down the function
- b is the rate of change (affects the horizontal stretch of the function)
- x is the time intervals

One possible approach to find the function that fits the model is through linearizing the function by taking the natural log of each term, since determining the function of a non-linear can be tricky. This method is called "linear regression", which is basically a method to determine the relationship between two variables by fitting a linear function to the data.[4] After converting to a linear function, it is possible to find the gradient and y-intercept, which can be substituted and converted back in its exponential form to determine all the constants. With the function $y = a \times e^{bx}$, it becomes:

$$\log_e y = \log_e a + \log_e e^{bx}$$

Using rules of logarithm, this can be simplified further to:

$$\log_e y = \log_e a + bx$$

Since $\log_e x = \ln x$, the function can be written as:

$$\ln y = \ln a + bx$$

The function now resembles the equation of a line: $y = mx + c$, where

- $y = \ln y$ as the y-axis
- $c = \ln a$ as the y-intercept
- $m = b$ as the gradient
- $x = x$ as the x-axis

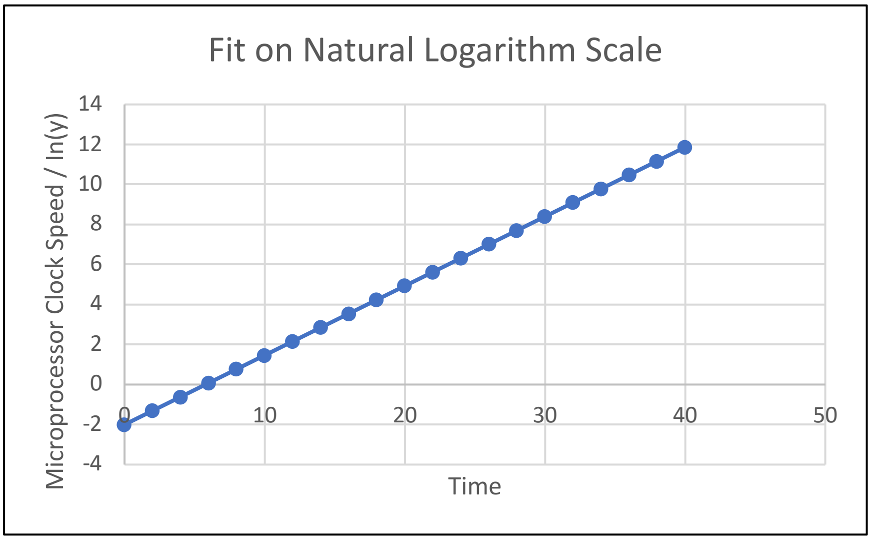
Taking the natural log of y, a new set of data can be generated:

Time (Year since 1976) (x)	Microprocessor Clock Speed (y)	ln(y)
0	0.135	-2.0024805
2	0.27	-1.3093333
4	0.54	-0.6161861
6	1.08	0.07696104
8	2.16	0.77010822
10	4.32	1.4632554
12	8.64	2.15640258
14	17.28	2.84954976

[4] Rod Pierce, "Least Squares Regression," March 2, 2019, https://www.mathsisfun.com/data/least-squares-regression.html)

16	34.56	3.54269694
18	69.12	4.23584412
20	138.24	4.92899131
22	276.48	5.62213849
24	552.96	6.31528567
26	1105.92	7.00843285
28	2211.84	7.70158003
30	4423.68	8.39472721
32	8847.36	9.08787439
34	17694.72	9.78102157
36	35389.44	10.4741687
38	70778.88	11.1673159
40	141557.76	11.8604631

Thus, a new graph in log scale can be produced:



This linear shape of the graph is expected since the increase is constant throughout at a doubling rate. The properties of this function are:

- The y-intercept is around -2, as can be observed from the graph.
- Finding the gradient of the function:

$$\frac{y_2 - y_1}{x_2 - x_1} = m$$

$$\frac{11.86 - 1.46}{40 - 10} = m$$

$$0.347 = m$$

- The formula for the log scale function is, therefore $\boldsymbol{y = 0.347x - 2}$

Since $y = 0.347x - 2$ is equivalent to $\ln y = bx + \ln a$, to convert it back to its corresponding exponential function, the logarithm law (stated below) can be applied.

$$a^x = b \leftrightarrow x \log_a b$$

$$\log_e y = bx + \log_e a \ [\log_e x = \ln x]$$

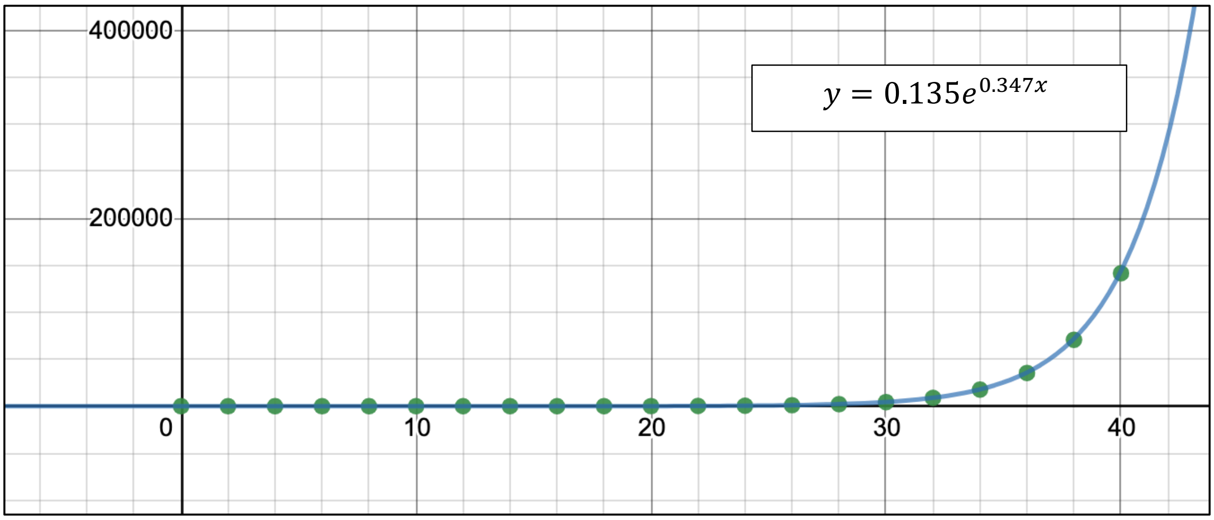
$$y = e^{bx} \times e^{\log_e a} \qquad [\log_e a = -2]\ [b = 0.347]$$

$$y = e^{-2} \times e^{0.347x}$$

Thus, the natural exponential function is now:

$$y = 0.135e^{0.347x}$$

Plotting the function against the model:



The function can describe the model perfectly. To double-check on the function, we can test the limit of the function against the limits stated earlier.

$$\lim_{x \to +\infty} (0.135e^{0.347x}) = 0.135e^{+\infty}$$

The limit to positive infinity will always return a positive value as the value e will always be positive.

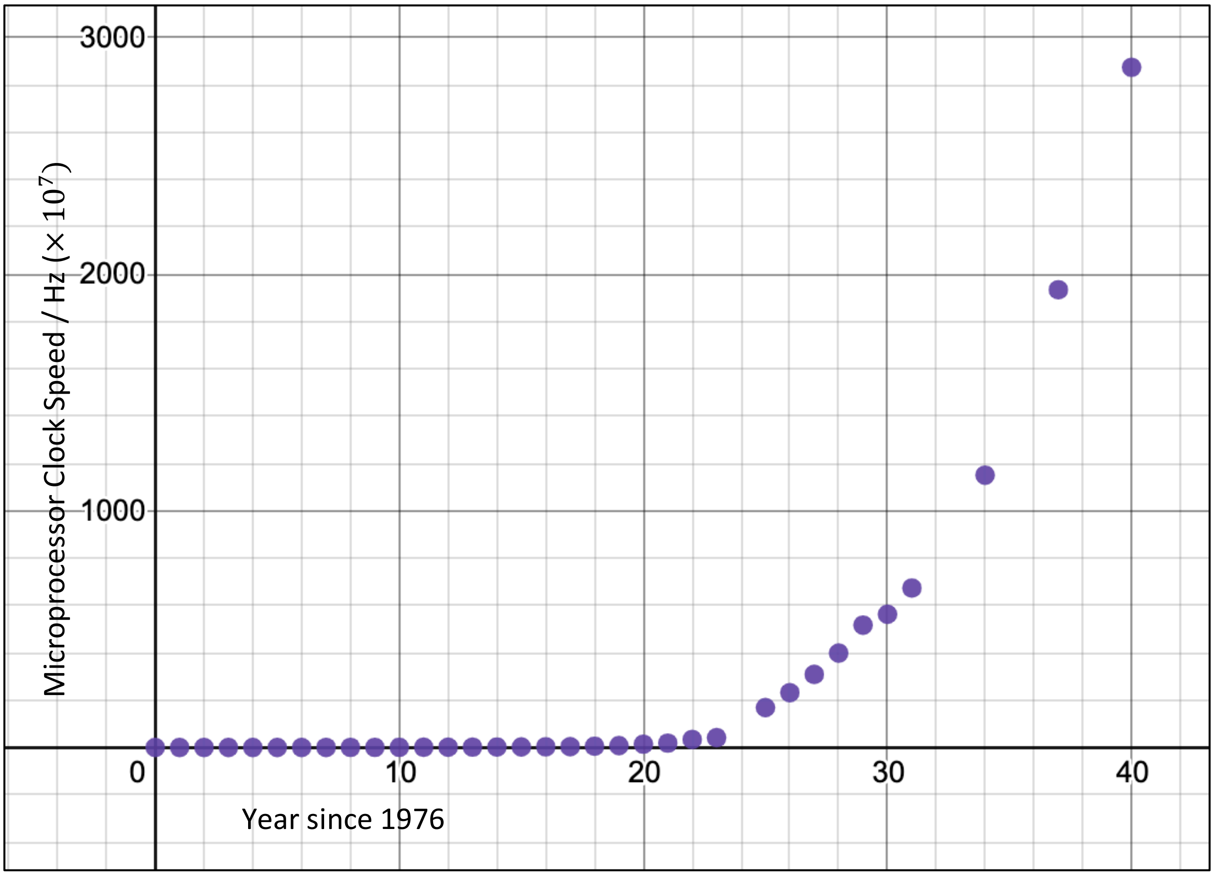
$$\lim_{x \to -\infty} (0.135e^{0.347x}) = 0.135e^{-\infty} = 0.135\ (0)$$

The limit to negative infinity will always return 0 as the value e as the x gets larger will becomes 0. This means that there is a limit at y = 0.

Both limit tests are successful, thus proving that the function describing the model is accurate.

Part 2: Actual Microprocessor's Clock Speed Model

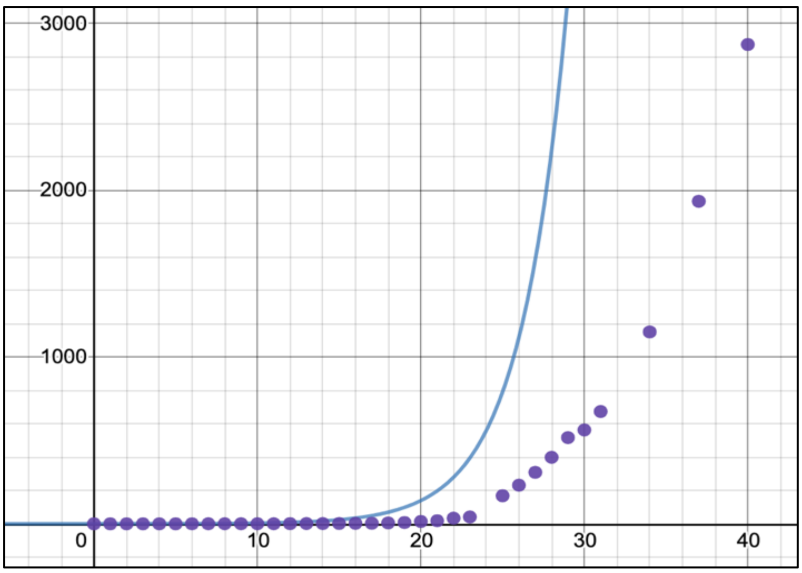
The data for the actual microprocessor's clock speed from 1976 to 2016 is attached in Table 1. Similarly, the values have been scaled down to $\times 10^7$, for the graph and values to be easily observable and modified.



From observing the model alone, it can be inferred that it is not a perfect exponential graphs there are several linear points between $x = 24$ and $x = 32$. Nonetheless, an exponential function will still work to describe the model as the anomalies only form a small part of the model.

Since the ideal function has been identified, it is possible to use it to compare with the real model to identify which variable(s) in the general function of exponential is likely to change:

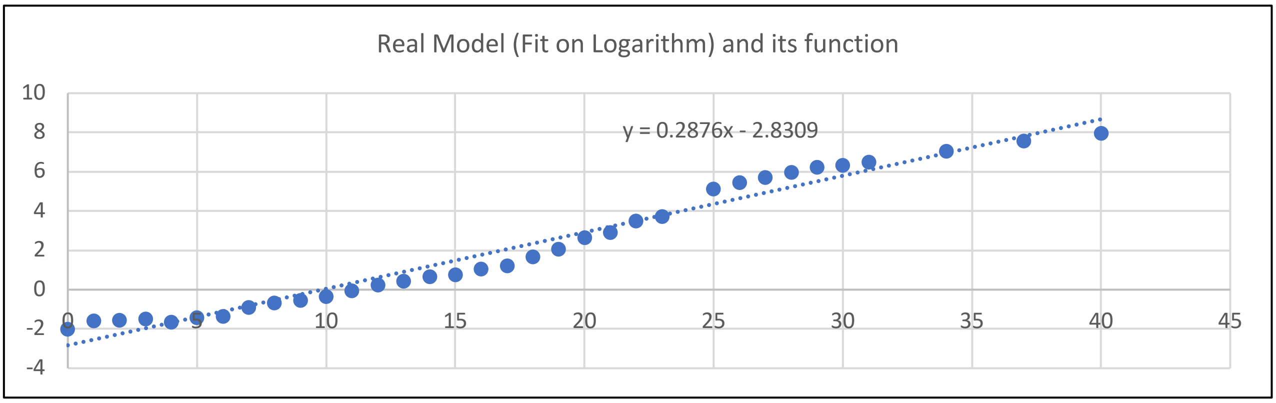
$$f(x) = a \times e^{bx-c} + d$$



Comparing both models, the following properties for the function for the real model can be deduced:

- There may be a horizontal stretch (affecting b), or/and
- There is a vertical stretch (affecting a),

Both will affect the curvature of the graph. It is unlikely that there is a horizontal or vertical translation since both models start at the same point. Basing on this assumption that there is no translation, it is possible to employ the same methods as earlier. By linearizing the data, then plotting it in terms of ln(x) against number of years, the gradient and intercept of the graph can be used to identify the exponential function that best describes the real model.



After algebraic manipulation to replace the ln into e^x, the function to describe the real model turns out to be:

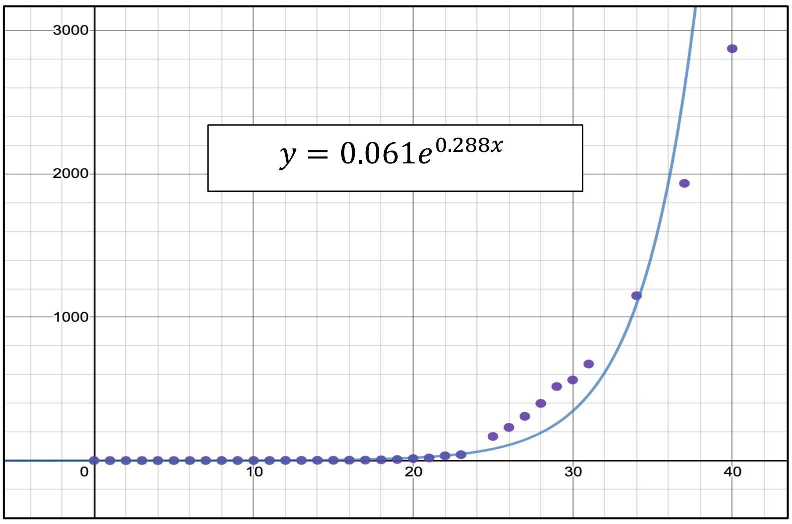
$$\log_e y = bx + \log_e a \text{ is equivalent to "} y = 0.2876x - 2.8309 \text{"}$$

$$y = e^{bx} \times e^{\log_e a} \quad [\log_e a = -2.8309]\ [b = 0.2876]$$

$$y = e^{-2.8309} \times e^{0.347x}$$

$$y = 0.061e^{0.288x}$$

This is plotted against the model, attached below:



The function, however, does not describe the model perfectly, as there are anomalies that do not fit in line with it. This is due to a possibility that during those periods, the average rate change increases faster or slower than the rest; it is faster when the points are above the line, while slower when they are below the line. These fluctuations limit the accuracy of the function, as the function would only provide a line of best fit that is in between all the data points. Nevertheless, the function still manages to describe the majority of the model to a good extent in general.

Part 3: Comparing Rate of Change

Now that both functions have been identified, the comparison between the average rate of change of the two functions can be found.

Ideal function: $y = 0.135e^{0.347x}$ Real function: $y = 0.061e^{0.288x}$

One possible way is through the first derivative of the functions, which can be found using the derivative of e^x:

$$\frac{dy}{dx} = e^{f(x)} \times f'(x).$$

For instance, the first derivative of the ideal function:

$$y = 0.135e^{0.347x}$$

$$\frac{dy}{dx} = 0.135 \times e^{0.347x} \times 0.347$$

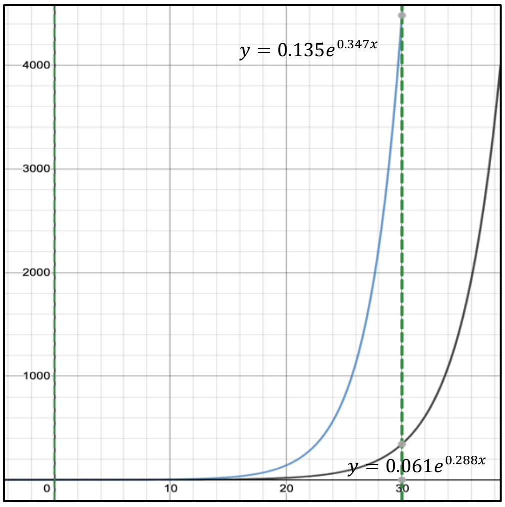
$$\frac{dy}{dx} = 0.047e^{0.347x}$$

Similarly, the first derivation of the real function is:

$$\frac{dy}{dx} = 0.018e^{0.288x}$$

From the derivates of both functions, it can be inferred that at any given value of x $\{x \subset \mathbb{R}\}$, the derivative of the real function is always smaller than the derivative of the ideal function. This means that the real function will always have a slower rate of change, hence gentler slope.

However, how much slower is the real function? This can be found through the average rate of change (gradient function): $\frac{y_2 - y_1}{x_2 - x_1}$. Through drawing a line connecting two points, the average rate of change between the 2 points can be calculated through the gradient function. To ensure the reliability, the same points of x have been chosen at $x = 0$, and $x = 30$.



Ideal Function		Real Function	
(0,0.135)	(30,4480)	(0,0.061)	(30,345)
$\frac{4480-0.135}{30-0}$ $= 149.3$		$\frac{345-0.061}{30-0}$ $= 11.5$	

The average rate of change of the real function is 137.8 Hz/year slower than the ideal function.

To conclude, Moore's Law is not able to perfectly describe the growth of microprocessor growth's speed. However, it is important to note that the law is accepted not for its applicability, but for its ability to predict the trend of computer growth. Despite not being able to describe the growth of microprocessor clock speed perfectly, the exponential growth of it thus fit the general prediction of Moore's Law, that technology grows in an exponential rate. This result is very applicable to the real-world situations, as this increase in technological power could be very beneficial to humankind. Especially with the exponential increase in microprocessor clock speed, computers could potentially operate faster than ever, thus allowing more complex calculations to be made in a shorter amount of time. This could be extremely useful for human to solve issues that are deemed to be impossible to right now, such as solving string theory in the Physics realm.

However, there could be several factors that may affect the validity of the result. For example, since the value is very large, the scale of the graph has to be minimized to a power of $\times 10^7$. Therefore, while the determined functions manage to describe both models to a great extent, it may not be able to describe them as well when the scale is increased back to its original. Plotting a graph without scaling it down is difficult to achieve due to the inability to observe the whole model clearly in one page.

One other improvement that can be made to enhance the exploration is to analyse the growth of non-computer related technology, such as commercial airplane flight speed, to figure out whether Moore's Law can perfectly describe the growth of other types of technology, which are also very applicable to the real world.

Works Cited

"Moore's Law." University of Missouri-St. Louis, May 17, 2013. http://www.umsl.edu/~siegelj/information_theory/projects/Bajramovic/www.umsl.edu/_abdcf/Cs4890/link1.html.

Pierce, Rod. "Least Squares Regression," March 2, 2019. https://www.mathsisfun.com/data/least-squares-regression.html.

Roser, Max, and Hannah Ritchie. "Technological Progress." Our World in Data, May 11, 2013. https://ourworldindata.org/technological-progress.

Rouse, Margaret. "What Is Clock Speed? - Definition from WhatIs.com." WhatIs.com, April 2005. https://whatis.techtarget.com/definition/clock-speed.