Computer Processor Power Draw Scaling with Voltage, Frequency and Temperature

Definitions of Key Terms and some Semiconductor Background Information:

Overclocking/Underclocking:
Overclocking is the act of increasing the operating frequency of a logic circuit in order to make it faster at the cost of increased power draw. Underclocking is lowering the operating frequency of a logic circuit to lower power draw at the cost of lower performance.

Transistor Properties:
A transistor is a semiconductor based device that works as a electrical switch. The difference between the two is that a switch requires kinetic energy to go from open to closed a transistor on the other hand only requires potential difference.

MOSFET:
MOSFET is a shortcut for Metal-Oxide-Semiconductor Field-Effect-Transistor and is the type of transistor used in all CPUs. When a MOSFET switches into the on state a capacitance is created causing the mosfet to draw current while it opens. When the MOSFET is switched off again the current drains into the ground. This property is responsible for most of the power consumption of semiconductor devices.

Leakage:
A quantum phenomenon where electrons and holes(places where electrons should be but aren't) jump from the source of a transistor into the insulator resulting in a small and constant current going through every transistor.

Manufacturing Process:
The manufacturing processes for CPUs these days operate on the nm scale and are very unreliable when it comes to leakage consistency resulting in no 2 CPU being exactly the same with one always being better or worse than the other in some aspects. This inconsistency is so high that even transistors in the same CPU are all different to a lesser or greater extent.

Voltage:
For the purposes of this paper the voltage being changed is only the core voltage. This is the voltage that the CPU cores and cache are subjected to. This voltage is usually between 1 and 2 volts for CPUs made after the year 2000 and is getting lower as CPUs get built on smaller manufacturing processes.

Cores:
The part of the CPU that does all computational work.

Intel Burn Test(IBT):
A program that uses all the CPUs available resources to their absolute maximum to test that they all function properly and to achieve a CPUs maximum power draw.

HWbot Prime:
A program that uses the CPU to hunt prime numbers. It uses a lot of CPU resources causing a high but not maximal power draw.

Abreviations

CPU: Abbreviation for central processing unit

GPU: Abbreviation for graphical processing unit

TDP: Thermal design power. A figure for the maximum heat output of a CPU given by the manufacturer

Abstract
The purpose of this study was to create an equation that would allow computer enthusiats to predict the power draw of a processor after changing it's operating voltage frequency and temperature. This will allow computer enthusiasts to better estimate the lifetime of a CPU at certain settings as well as the heat output and strain put on the power delivery components within the PC.

To do this I tested an AMD CPU in many different configurations and analyzed the results of these tests for linear and nonlinear relationships.

My results were conclusive for my CPU and allow me to predict it's power draw well within a 5% margin. However this is only one CPU and not all CPUs are the same. Some may scale differently and due to limitations in testing equipment even for my own CPU there are some configurations that I can not predict due to a lack of data greatly impacting the analysis. More testing with better equipment will be able to complete the equation but my data is insufficient for doing more than spotting trends.

Table of contents:

Part 1: Introduction and Purpose of Study

The goal of this research is to create a formula that will give overclocked and under clocked power draw for any semiconductor device given the factory power draw, voltage and frequency at the intended operating temperature of the overclock or underclock with a ± 5% accuracy. This formula will allow anyone picking computer components or designing them to easily predict if they need a better power source or more cooling. For example may current Nvidia GPUs suffer from having power deliver circuitry that is not on par with the power draw capabilities of the GPUs resulting in some people burning out cards when overclocking by applying high core voltages(1.4V to 1.5V) for short term(less than 24 hour) performance boosts. With this equation people will be able to predict if they need to buy a GPU with a custom power delivery design or not. The same goes for motherboards intended to run overclocked CPUs that draw over 125W at factory settings.

Processor power draw is a major issue in today's electronics. Before 2000 a desktop processors pulled as little as 8W and was 294mm^2 so cooling and power delivery circuitry was of little concern when designing a computer. However today your average CPU pulls 54W and is 177mm^2 and the most powerful CPUs run on 130 to 220W with die sizes of 257mm^2 to 315mm^2. Graphics processing units pull even more power with most high end cards running on over 200W with some rare cases pulling over 300W. Most of these components can be overclocked to run faster at the cost of more power draw or underclocked to run slower on less power. Temperature also affects the electrical properties of many materials and can also impact the power draw of semiconductors

The Significance of TDP

The very first thing to test was how much power the CPU pulls at completely default setting in relation it's factory power rating because the factory power rating and voltage is all the information the owner of this CPU has without doing complicated measuring. The default voltage for this CPU turned out to be 1.422V while running Intel Burn Test. To make sure that the CPU wasn't "cheating" the power draw test by choosing to run at lower than specified frequencies I manually set the CPU to run at 4Ghz and disabled all power saving features. The measured current draw for this configuration was 4.37A which is 52.44W which about 80% of what the CPU is rated at. When conducting this test I chose to use the heatsink that came with the CPU and did not control the temperature that the CPU was at while testing because the TDP is what the CPU is supposed to draw in it's default configuration. Which meant using the default cooler. However for whatever reason the CPU never got past 45C° which is about 20C° below intended operating temperature. From later testing it is obvious that the low temperature caused the low power draw.

Method

For testing I used a Gigabyte A88X-D3H motherboard. To measure the current going to the CPU I took a EPS 8pin extension cable from AKASA and cut it up halfway through the +12V cables(the yellow ones). I then connected only 3 of these 4 cables to 3 digital multimeter using screw terminals. The 4th cable was left disconnected because the multimeter I wanted to use on it wasn't working. I connected this cable to the motherboard and the other end to the 8pin EPS of an EVGA 430W power supply. I used 1 stick of generic 1.5V 1600mhz 9-9-9-24 1T 4GB RAM. To check temperature I used a thermometer with a thermocouple capable of measuring down to -140C. The thermocouple was placed inside in the small hole at the base of the de8auer ECC Fusion rev 3.1 LN2 CPU container I used for cooling. To make sure that the thermocouple made contact with the container I used Arctic MX-2 thermal paste. The container was attached to the motherboard using the supplied mounting mechanism and I used Arctic MX-2 to get good contact between the base of the pot and the CPU. To make sure I was getting accurate voltage readings I soldered a solid copper wire to the positive leg of one of the output capacitors of the VRM this wire was connected to a digital multimeter using a alligator clip. The other lead of the multimeter was stuck into the common ground contact of a molex 4 pin cable coming out of the PSU.

All the following results are from tests using a load lower than that of IBT in the form of HWbot Prime. I used HWbot Prime because it has a more consistent and lower load on CPU resulting in more stable power readings and less temperature regulation issues.

To allow the upscaling of all readings below I also tested the power draw of the CPU with same settings as IBT but running HWbot Prime. This resulted in a current of 3.16A which means that HWbot Prime is about 72.3% of the load of IBT.

To control temperatures I poured small amounts of liquid nitrogen into the pot and waited for the temperature to go down to 1 or 2C° below my target temperature. I would then start HWbot prime and wait until the thermometer showed the temperature I was waiting for at which point I would take the reading.

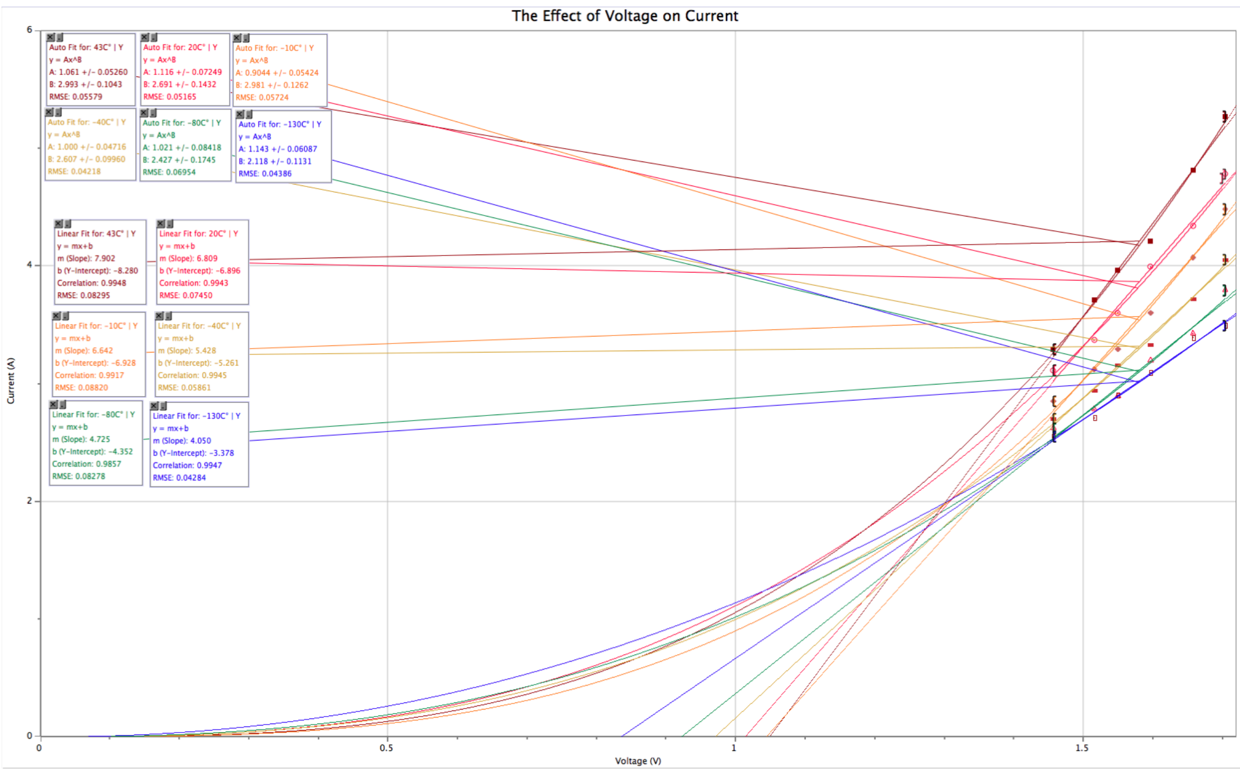
Test Results

The Effects of Voltage and Temperature

Changing the core voltage of a CPU has the most significant effects on the CPU's power draw as it dictates the increases in current and leakage.

Below is a graph of the currents draw at different voltages and temperatures.

Fig 1 the effect oof voltage on current

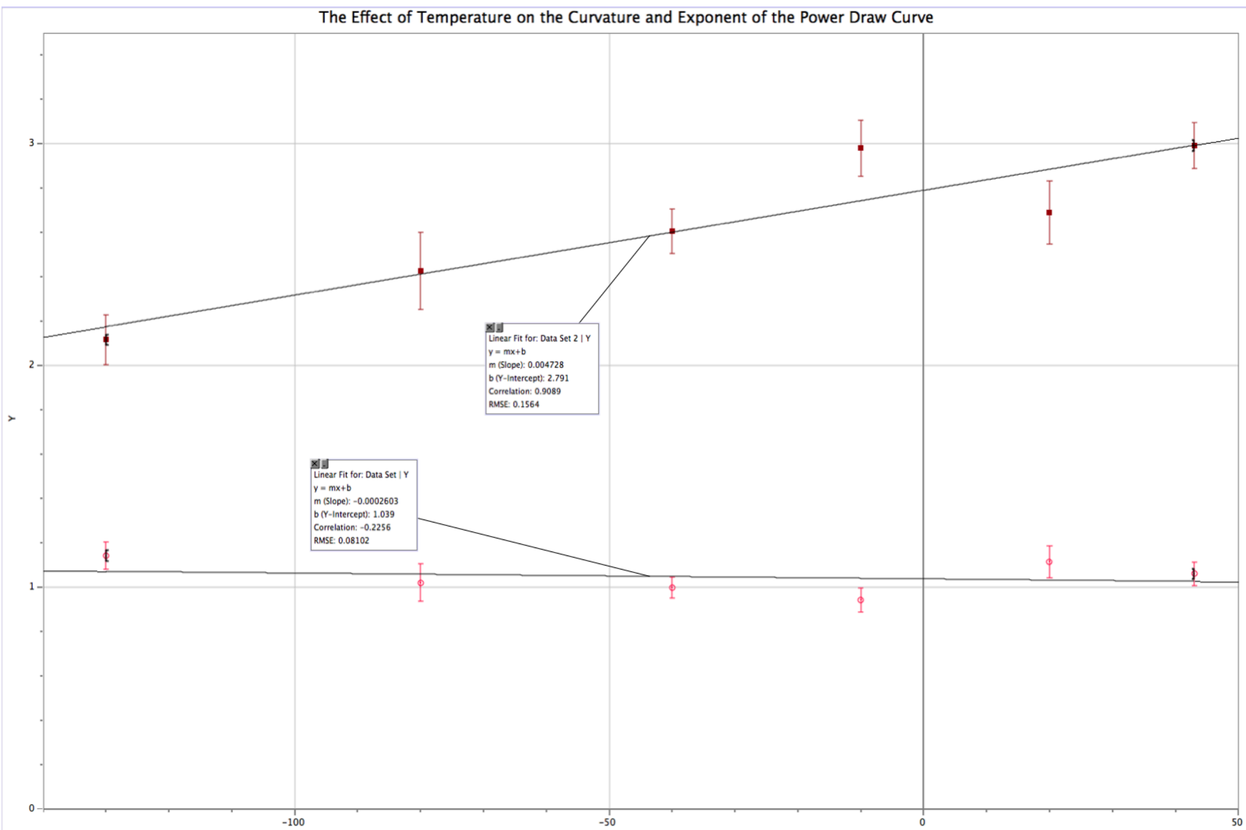


The curves in Fig V1: **The lines in Fig V1:**

43C°	$y=1.061x^{2.993}$	43C°	y=7.902x-8.280
20C°	$y=1.116x^{2.691}$	20C°	y=6.809x-6.896
-10C°	$y=0.9044x^{2.981}$	-10C°	y=6.642x-6.928
-40°	$y=x^{2.607}$	-40°	y=5.428x-5.261
-80C°	$y=1.021x^{2.427}$	-80C°	y=4.725x-4.352
-130C°	$y=1.143x^{2.118}$	-130C°	y=4.050x-3.378

The scaling of current with voltage can be either linear or a curve because equipment limitations did not allow the testing of voltages below 1.4V. A curve fits the data ever so slightly better than a line however this may be completely coincidental and does not account for the fact that below a given voltage a transistor will simply not activate. A line satisfies the inability for a transistor to activate below a certain voltage however a decrease in temperature causes transistor's electrical resistance to increase and as such lower temperatures should have a higher cutoff voltage which is not satisfied by either model as such further testing would be required to see the activation voltage cutoff points.

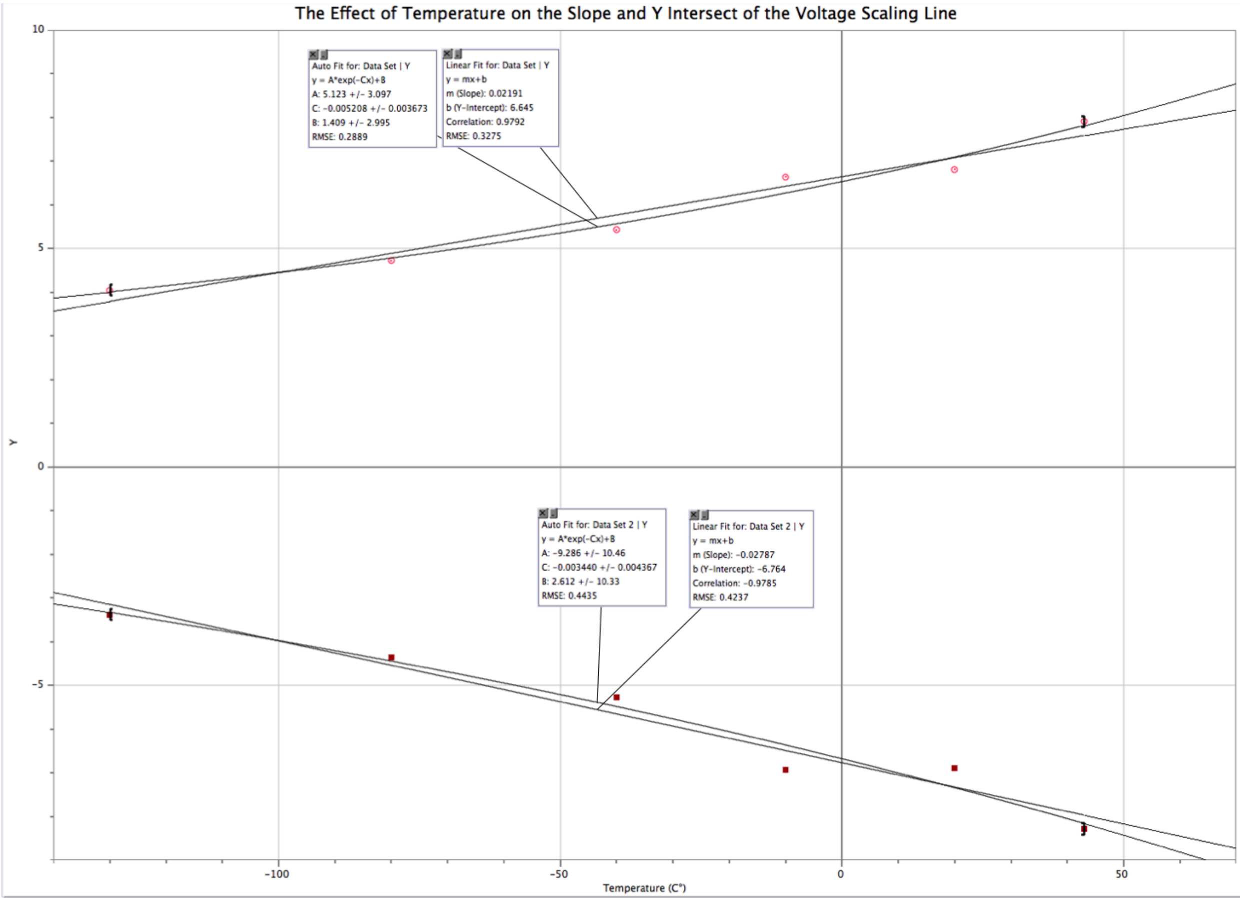
Fig 2 The effect of temperature on the A and B values of the Ax^B model from Fig 1



Analyzing the constants of the curved and linear models of current draw gives the graphs in Fig 2. The equations of the lines in the first grah allow the predicting of the shape of the voltage scaling curve for the temperature X. The lower line is the A value and the upper line is the B value of the Ax^B model. The error bars are the error margins that logger pro gave for the A and B values when it approximated them.

In further analysis I will only be using the linear model for voltage scaling because the data I have supports both the curved and the linear model but the linear model is easier to manipulate and use in calculations.

Fig 3 the effect temperature on the Y intercept and slope of the lines in Fig 1

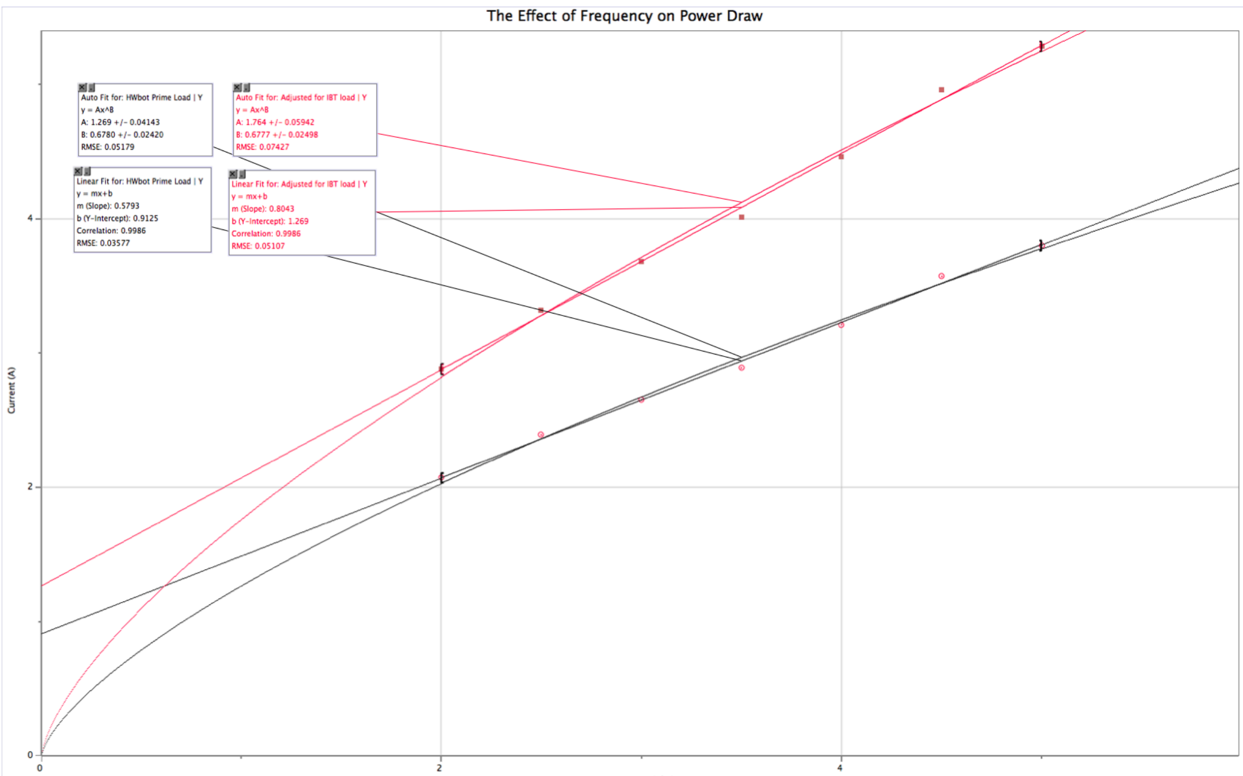


The points above 0 give in Fig V3 are the slopes of given temperatures and the points below zero are the Y intercepts. Both seem to be scaling almost linearly but fit the exponential curve of A*exp(-Cx)+B ever so slightly better because it passes much closer to most of the values than the best fit line.

The effect of frequency

Frequency scaling testing results also had a large range when comparing the percent change in power draw to the last used setting however the range was low when comparing the percent change in power draw compared to the original setting of 2Ghz. The trend of increasing efficiency shows up however it is not as significant as in the externally sourced data. Sub 1Ghz testing is needed to disprove the existence of a linear relationship between current draw and frequency if I didn't have externally sourced data from sin0822 that showed this trend more clearly.

Fig 4



The curve in Fig 4 is formed by the equation Ax^B where A is 1.269 and B is 0.678 and x is the frequency the CPU is operating at. Frequency scales mostly linearly with frequency the slope of this line is different between CPUs and it also differs based on the range of the data used to model it and the load applied to the CPU when gathering the data. Therefore it is impossible to get the frequency curve or line accurately without doing testing for every CPU model separately.

Therefore one possible method of predicting the curve is to assume a linear relationship and apply it to TDP to get the power draw of one higher frequency than stock and then based on these 2 points predict the curve. The slope used in these calculations would be 0.81 per 1% change in frequency since that is the average of the slope of sin0822's data and mine. Here is an example of doing this for the CPU I used:

The CPU has a 65W TDP
Therefore it has a current draw of 65W÷12V= 5.41A
Its stock frequency is 4Ghz
The predicted of 5Ghz is 5Ghz/4Ghz = 125%
25% * 0.81 = 20.25%
120.25% of 65W is about 78.16W which is 6.51A

This calculation is predicting the current draw at room temperature if however, you substitute the TDP value for the current draw value obtained at 4Ghz in testing(4.46A IBT adjusted) and multiply it by 120.25% as in the example then you get 5.36A which is 1.5% of from the measured value. This amount of error is acceptable because when using the people use this calculation they can just gve themselves a 5% overhead for safety without being wasteful of available resources. However this amount of error does not allow for getting a current curve for frequency because if you graph these values and try a Ax^B curve fit on them you get an A of 1.424 and a B of 0.8238 whereas the IBT adjusted measured values get a curve with an A of 1.764 and B of 0.6777 that's a greater than 20% difference in A and B values. Which is way too much error to be acceptable.

The Formulating of the Equation and the Conclusion

Voltage scaling

Due to budget limitations for running test I can not determine some things from my voltage scaling graphs. These are whether or not scaling is linear or a curve even though I assume it to be a curve on the basis that W=IV and I = V/R so if R is to remain constant we get that W=(V/R)*V that's $W=V^2/R$. A constant R means a quadratic relationship between voltage and power draw. However my results show that it is not the case as in that situation all the voltage curves would have been Ax^2 where x is voltage. As such I believe that AMD CPUs do not retain a constant resistance as the operating voltage increases and instead the resistance changes with the voltage due to some transistors in the CPU reacting to the higher voltage differently resulting in a more open channel in the same amount of time.

Taking this into account voltage scaling above 1.45V being curved or linear is not all that different and as such a linear scaling model even if wrong can prove useful for high voltages on 32nm AMD CPUs.

For voltages above 1.45V calculating power draw is a simple case of taking the target operating temperature and using it to calculate the slope and Y intercept of the voltage line for said temperature using either the curved or linear models from the graph tracking the effects of temperature on A and B values. I've decided that the curved model is better because 2 points of data fit it perfectly and the other points fit it about as well as the linear model however I would still like to gather more data to fully prove one of the 2 models as correct. However budget constraints got in the way of doing further testing.

These models are $M = 5.123e^{0.005208x}+1.409$ for the slope and $B=-9.286e^{0.00344x}+2.612$ for the Y intercept. So for predicting the power draw of the Athlon X2 370K a person would just replace the x in both of those equations with the temperature at which they will allow their CPU to operate and they will get the values for A = Mx+B where A is current and x is the voltage the owner plans to use for their CPU. The result of that equation is merely the current pulled from the PSU so to calculate the power draw in W the result needs to be multiplied by 12. To get the amount of current that the VRM is having to supply the user would multiply by 12 and divide by the voltage they plan to use. If this result is greater than the sum of the current ratings of the MOSFETs the motherboard uses the MOSFETs will burn up under load.

The issue with the equation is that it gives zero information on how the initial TDP relates to the resulting power draw because I haven't tested that however I assume that differences in TDP of of the model can be made up by adding the following to the linear model: $A=Mx+B(TDP_2 / 65)$. This adjustment will lift the entire model to a new starting power level and as such should allow the factoring in of TDP differences.

Frequency scaling

Frequency scaling proves to work very well with both the external data and my own data showing similar trends. On average a 1% increase in frequency causes an increase in power draw which is about 0.81%. For common uses this linear relationship works well enough. However when frequencies are very very low the relationship breaks down and a better model would be a non linear one. My testing nor the external testing fully prove this prediction however they do hint at it being very possible. Further testing is needed to prove a curved model however for the vast majority of overclockers the linear relationship of 1% more voltage = 0.81% more power draw is good enough.

The equation for frequency scaling is:

$((((F_2/F_1-1)*100)*0.81+1)+100)/100$ where F_1 is the stock frequency and F_2 the overclocked or underclocked frequency.

Which can be shortened to this:

$((F_2/F_1-1)*81+101)/100$

Putting it together

From the above a general model would look like this:

$$((F_2/F_1-1)*81+1)/100)*((5.123e^{-0.005208(T)}+1.409)V+(-9.286e^{-0.00344(T)}+2.612))$$

Where F_1 is the frequency of the CPU used for voltage scaling testing so 4.2 in this case. The F_2 is the predicted overclocked frequency. The V is the predicted voltage and the T is the maximum temperature the user allows

So with all this information about scaling how exactly would one calculate the power draw of a CPU. First of all the voltage scaling testing was done at 4.2Ghz and as such all frequency scaling has to be calculated by taking the % difference between the target frequency and 4.2Ghz. For voltage the user has to pick a temperature and then a voltage so going with that logic the maximum recommended daily use safe temperature for an AMD CPU is 70C° however I can only compare to the power draw at 46C° so I'll use that. So a 4.5 Ghz overclock running at 1.5V would use:

$$(((4.5/4.2-1)*81+101)/100)*((5.123e^{0.0052(46)}+1.409)1.5+(-9.286e^{0.0034(46)}+2.612))=1.067*((7.92)1.504+(-8.25))= 3.91A$$

My measured value for 4.5Ghz at 46C and 1.504V was 3.89A so the equation is of by only 20mA which is 0.5% well within error margin target I set for myself. However without further testing I will never know if it works for voltages below 1.45V where it might undershoot and turn out to do more harm than by causing people to design things with lower maximum power expectations and causing things to burn out when the real values exceed the values predicted by the equation.

Reflection

When I first started overclocking computers I theorized that power draw should be predictable using this equation: $W_{oc} = (V_{oc}^2/(V_s^2/TDP))*(F_{oc}/F_s)$. This equation is used by several websites meant to help people choosing their power supply. This paper was supposed to find an equation that worked for all CPUs. Unfortunately the paper didn't achieve that because when I was doing my TDP testing didn't show that TDP = Power draw. Now I strongly believe that is entirely because I was testing at too low a temperature but I didn't even have the equipment needed to keep the CPU at a steady 70C while running Intel Burn Test so I couldn't do that. I could have used my temperature scaling data to correct for this however that would have meant that too much of my data was calculated values.

If I was to do this again I would have gotten more CPUs to test with. Better motherboards because the one I used here didn't allow voltage below 1.4V which severely impacted my voltage scaling data. I would also would have liked more liquid nitrogen as that would allow me to test more temperatures before running out of coolant.

In the end I did not achieve my goal. I do not have a general model that always works however I did get close and with just a few refinements to the testing method and sample sizes I would hopefully be able to produce a fully functional equation.

Bibliography:

Sin0822. "**Ivy Bridge Overclocking Guide** (with LN2 Guide at The End)." *Overclock.net*. Enthusiast, 23 4 2012. Web. 4 May 2014. <http://www.overclock.net/t/1247413/ivy-bridge-overclocking-guide-with-ln2-guide-at-the-end>.

Stevenvh. "Power consumption is about linear with frequency." *Stack Exchange Physics*. Stack Exchange, 23 8 2012. Web. 4 May 2014. <http://physics.stackexchange.com/questions/34766/how-does-power-consumption-vary-with-the-processor-frequency-in-a-typical-comput>.

Hartung, Roman. "The Risks of Overclocking." *Overclocking.Guide*. Der8auer, 7 Oct. 2014. Web. 23 Oct. 2014.

Appendix

Raw data for sin0822's frequency tests and efficiency figures.

CPU Frequency @ 1.4V i7 3770K						
Mhz	Power Draw (W)	% change in power draw vs 4Ghz	% change in frequency vs 4Ghz	Frequency Ratio to Power Ratio	Efficiency Mhz / W	Observations
5000	159	20.45	25.00	1.22	31.45	A small increase(upto 300mhz) in frequency is of lower efficiency than a large increase(equal or greater than 400mhz). The boost in efficiency by maxing frequency is: 3.8%
4900	158	19.70	22.50	1.14	31.01	
4800	154	16.67	20.00	1.20	31.17	
4700	153	15.91	17.50	1.10	30.72	
4600	150	13.64	15.00	1.10	30.67	
4500	147	11.36	12.50	1.10	30.61	
4400	144	9.09	10.00	1.10	30.56	
4300	142	7.58	7.50	0.99	30.28	
4200	139	5.30	5.00	0.94	30.22	
4100	136	3.03	2.50	0.83	30.15	
4000	132	N/A	N/A	N/A	30.30	