Ryan Nakamoto

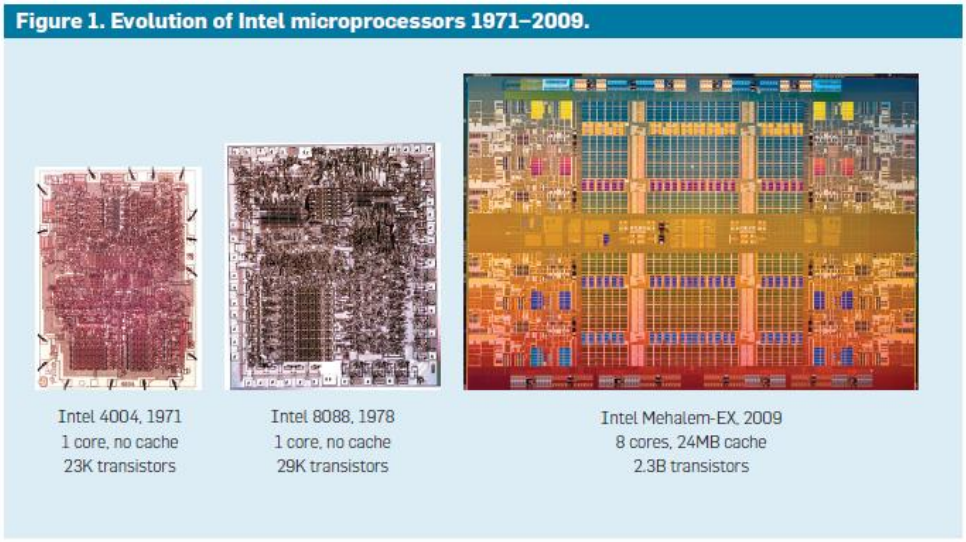
Dr. John Stewart

Universtiy Physics II

November 18th, 2011

Moore's Law

Introduction



(Borkar and Chien, The Future of Microprocessors 2011)

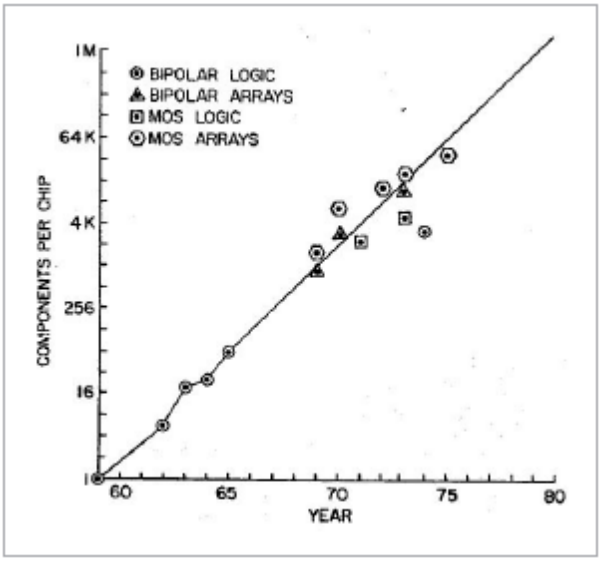
In 1965, Gordon Moore, co-founder of Intel, released his paper, *Cramming More Components onto Integrated Circuits*, which predicted that the transistor density on an integrated circuit would be able to double every year (Moore, Cramming More Components onto Integrated Circuits 1965). This estimate was refined to doubling every two years by Moore in 1975 (Moore, Progress in Digital Integrated Electronics 1975). These predictions proved to be so accurate it was dubbed Moore's Law, and has been the major driving force in computer technology; going

from 23,000 transistors in Intel's CPU in 1971 to billions of transistors in some modern CPUs, as shown in figure 1 (Borkar and Chien, The Future of Microprocessors 2011).

However, with limitations in the scaling of transistor count to actual processor speed; innovations, such as the multi-core processor, were made to keep the growth in computer technology up to speed (Borkar and Chien, The Future of Microprocessors 2011). Recently, power supply limitations, called Dark Silicon, force processors to only use 70 percent of their potential and this percentage is decreasing as the number of cores is pushed higher. This causes the speed gains due to cores to decrease greatly, leaving a 24-fold gap between the Moore's laws predictions, and current estimate of processing speeds in 2024 (Esmaeilzadeh, et al. 2011). Luckily, research on increasing the energy efficiency of processors and load-times of programs being processed by them; hope to alleviate the strain on power supplies, while pushing the processing speed even further (Sampson, et al. 2011).

Background

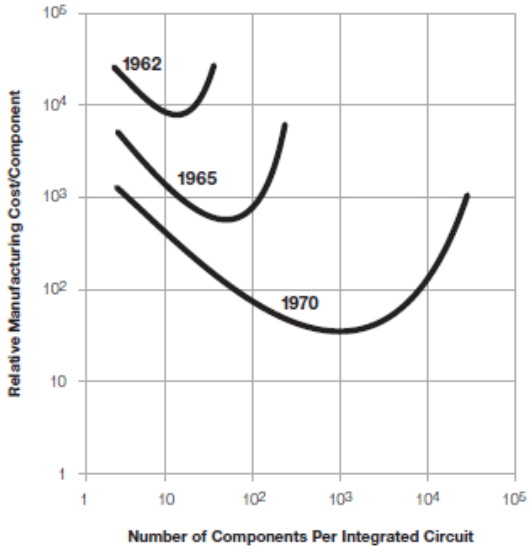
Figure 2: Moore's Law Curve



(Moore, Progress in Digital Integrated Electronics 1975)

Moore's Law, as refined in 1975, states that transistor density on an integrated circuit will double every two years (Moore, Progress in Digital Integrated Electronics 1975). At first, this was achieved through Dennard scaling; which involved reducing oxide thickness, transistor length, and width by a constant factor (1/k) every generation (Kuhn 2009). This allowed the area of transistors to drop by 50 percent every 2 years, while maintaining a constant electric field. This allowed the processor performance to increase by 40 percent every year as shown in figure 2; either through a 30 percent reduction in the delay of processes, or a 40 percent increase in processing speed. While at the same time this allowed for a decrease in power per transistor by 50 percent, and with each generation having twice the transistors meant a constant energy demand.

Figure 3: Component Cost Curve



(Moore, Cramming More Components onto Integrated Circuits 1965)

At the backbone of this law was what many consider to be Moore's Second Law. He theorized that as the number of components per circuit increased, the minimum cost per component would decrease at nearly the same rate, as shown in figure 3. This would allow for electronics companies to double the number of components per circuit while keeping the costs of production at a constant rate. This would allow the production and advancement of microprocessor technology to become extremely economically viable. However, with only a 40 percent increase in processing speed, Moore's Law would see diminishing returns; which would cause a need for further innovation to keep improvements coming at a steady rate (Borkar and Chien, The Future of Microprocessors 2011).

Multi-core processors are just what were needed to fill that gap. By using Pollack's Rule, the increase in performance is proportional to the square root of the increase in complexity; two smaller cores can cause a boost of 70 to 80 percent compared to a similarly sized single-core processor. In addition, individual cores of multi-core processors can be turned on and off individually, saving power for the system. Also, work can be separated between the cores; both heat and work-load can be distributed evenly across both of them. However, this is not without a few drawbacks (Borkar, Thousand Core Chips- A Technology Perspective 2007).

Due to the nature of multi-core processors, programs must be specialized with multiple commands that can be run simultaneously for each core, or threads. Before, during the popularity of single core processors, programs shared in the additional improvement without any extra input on their part, because a 40 percent increase in processing speed meant the program would just be run 40 percent faster. However, with parallel processing, the program must be specially tailored to take advantage of having different cores running threads simultaneously, or it will not see the full gain in performance. Creating programs that can utilize the multiple threads properly takes a

lot of time and resources, and is sometimes just inappropriate for certain programs. This means that an improvement in processing speed does not necessarily create a one to one increase in computer speed. Luckily most CPU's have a slightly larger core for handling large single-thread workloads (Sutter 2005).

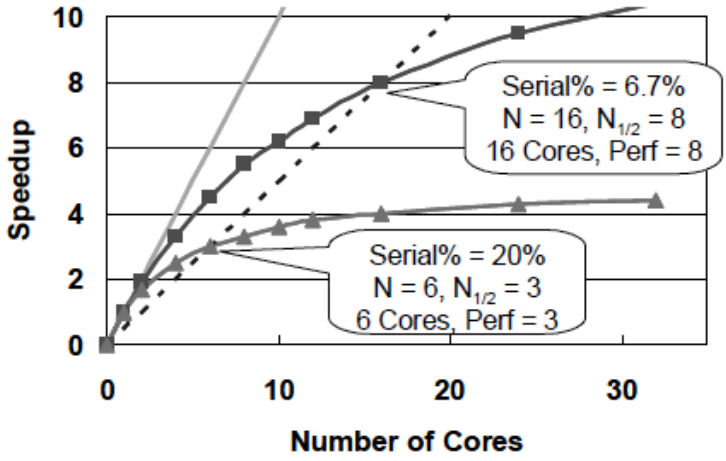


Figure 4: Amdahl's Law Limiting Cores

(Borkar, Thousand Core Chips- A Technology Perspective 2007)

In addition, adding additional cores comes with an even greater diminishing return. According to Amdahl's Law, where the increase in parallel processing speeds, *S*, is diminished by the percentage of single threaded code in a program, *P*.

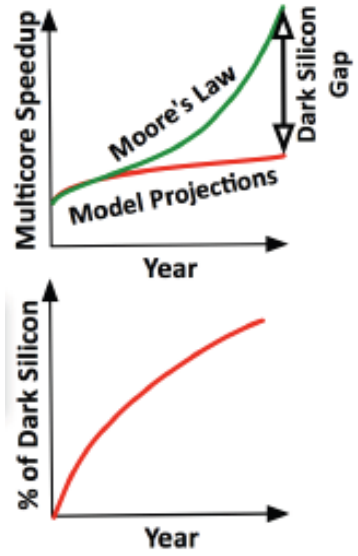
$$S = \frac{1}{(P + \frac{1-P}{N})}$$

where *N* is the number of cores in the processor. So, even with a percentage of single threaded codes as small as seven percent you will see an adverse diminishing return beyond 16 cores, as shown in figure 4. Also, with attempting to run multiple cores simultaneously, power demands become unreasonable. This requires designers to consider how to run the cores efficiently enough to be able to maintain a 100W limit (Borkar, Thousand Core Chips- A Technology Perspective 2007).

Fine grain power management helps to alleviate most of the power problems with processors. It allows the processor to be run at varying specific frequencies, tailored for the amount of processing power it actually needs. As technology advances, the number and complexity of power settings for the processors will increase. For example, before this most processors only had two settings: on and off. (Borkar, Thousand Core Chips- A Technology Perspective 2007). Now with the introduction of 10 by 10 optimization, processor frequencies can be throttled in chunks of 10 percent in order to make the processor extremely energy efficient (Borkar and Chien, The Future of Microprocessors 2011).

Figure 5: Dark Silicon

(Esmaeilzadeh, et al. 2011)



As the number of cores in processors begins to climb, researchers are starting to find an effect called Dark Silicon. With each core introduced to a processor, the percentage of the processor that can be used at full capacity drops exponentially, this is called the utilization wall. This leaves projections showing the speed increase only increasing by 7.9 times by 2024. This will leave a 24-fold gap between current projections and the expected growth rate by Moore's Law, as shown in figure 5. These problems show what may be the end to Moore's Law, as processing performance rates may begin to plateau (Esmaeilzadeh, et al. 2011).

Recent research has begun to create high efficiency cores to combat the problems stated in Dark Silicon. The major result of dark silicon is that the opportunity cost of creating highly specialized processors is decreasing as the number of cores increase. So, by combining large numbers of operations and complex data strings into "fat" operators, this helps to minimize the path length and load times for operations. In addition, changing clock speeds for certain

processes helps to maximize efficiency as well. For example, allowing memory requests to operate at a higher clock speed than data saves power at the same time as making the processor faster overall by increasing the memory speed of the processor itself. The application of these techniques can lead to a 35 percent savings in power usage, and a 150% increase in speed of the processor itself (Sampson, et al. 2011).

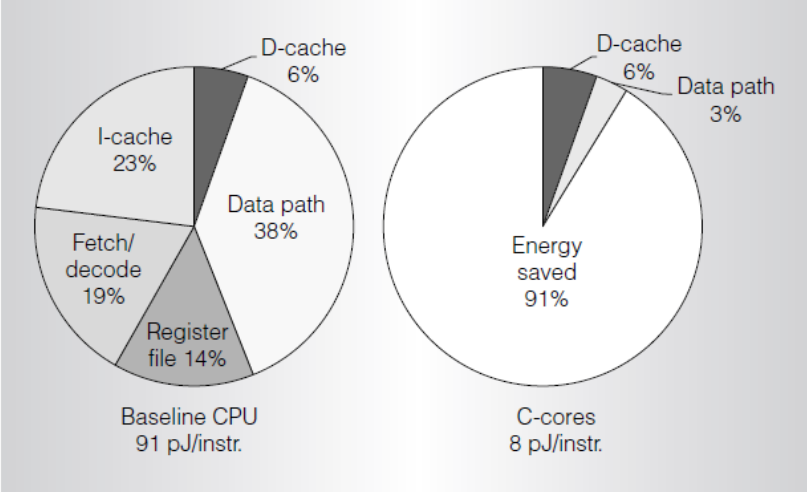


Figure 6: Energy Savings by C-Cores

(Goulding-Hotta, et al. 2011)

Due to the even greater efficiency demands of today's mobile phones; research in finding ways around the utilization wall presented in Dark Silicon is pivotal, as phones become more and more powerful. Especially with phones having an extremely low power budget, only 1 percent of a 32nm processor can be used at full frequency. The separation of codes into two branches: hot and cold, allow for the low energy requiring code, or cold code, to be processed by the standard CPU. While using high efficiency cores (C-cores) process the more intensive code strings (hot code). This method uses 18 times less energy than standard processors on mobile phones, as shown in figure 6 (Goulding-Hotta, et al. 2011).

Unfortunately, while CPU speed has seen enormous gains in the past decades, certain hardware components have failed to keep up. This creates a bottle-necking effect where the CPU is waiting on the other components to catch up. That causes an overall decrease in performance, and a new demand for higher performing hardware to counteract the delay. For example, on hard drive's disc access speed causes latency for the CPU, so the popularity of solid state drives has increased due to their higher average access speed (Moore's Law 2011).

In addition, new materials called strains have been added to transistors to increase their efficiency. Such as HiK dielectrics used to create greater power efficiency and transistor scaling in order to improve the performance of processors. However, it is not without its problems, HiK dielectrics tend to overheat faster making them harder to use as the number of processing cores. So far the current strains used to fabricate today's processors (e-SiGe or e-SiC) will continue to drive at least basic Dennard scaling forward to see increasing processing gains. Unfortunately, the most interesting form of fabrication of processors is using a lattice-mismatched to silicon, which would require a total overhaul of how we scale processors today. However, this means that the research and development of new materials will also continue to drive processing technology and speed forward (Kuhn 2009).

Conclusion

Moore's Law, a prediction that has driven the computer technology industry for the past 3 decades is seeing a potential end. Hopefully, new research in power efficiency, like that shown in the new c- cores, and multithreading codes for processing technology will allow Moore's Law

to continue. If not, we may sadly see the plateauing of processors, and by association most technological innovations that require higher processing speeds. Likely however, it will only mean a gradual slowing of innovation rather than a halt all together.

Works Cited

1. Moore, Gordon. "Cramming More Components onto Integrated Circuits."*Electronics*. 38. no. 8 (1965). download.intel.com/museum/Moores_Law/Articles-Press_Releases/Gordon_Moore_1965_Article.pdf (accessed November 26, 2011).
2. Moore, Gordon. "Progress in Digital Integrated Electronics." *Electron Devices Meeting*. 21. (1975): 11-13. http://0-ieeexplore.ieee.org.library.uark.edu/stamp/stamp.jsp?tp=&arnumber=4804410 (accessed November 26, 2011).
3. Kuhn, Kelin. Intel, "Moore's Law After 32nm: The Challenges in Physics and Technology Scaling." Last modified 2009. Accessed November 26, 2011. http://download.intel.com/pressroom/pdf/kkuhn/Kuhn_SSDM_plenary_text.pdf.
4. Sutter, Herb. "The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software." *Dr. Dobb's Journal*. 30. no. 3 (2005). https://docs.google.com/viewer?url=http://www.info2.uqam.ca/~tremblay/INF5171/Liens/sutter.pdf (accessed November 26, 2011).
5. Esmaeilzadeh, Hadi, Emily Blem, Renée St. Amant, Karthikeyan Sankaralingam, and Doug Burger. International Symposium on Computer Architecture, "Dark Silicon and the End of Multicore Scaling." Last modified June, 2011. https://docs.google.com/viewer?url=http://research.cs.wisc.edu/vertical/papers/2011/isca11-darksilicon.pdf.
6. Sampson, Jack, Ganesh Venkatesh, Nathan Goulding-Hotta, Saturnino Garcia, Steven Swanson, and Michael Taylor. "Efficient complex operators for irregular codes." (2011).

http://0-ieeexplore.ieee.org.library.uark.edu/stamp/stamp.jsp?tp=&arnumber=5749754 (accessed November 28, 2011).

7. Borkar, Shekhar. Intel, "Thousand Core Chips-A Technology Perspective." Last modified 2007. http://videos.dac.com/44th/papers/42_1.pdf.
8. Goulding-Hotta, Nathan, Jack Sampson, Ganesh Venkatesh, Saturnino Garcia, Joe Auricchio, Po-Chao Huang, Manish Arora, Siddhartha Nath, Vikram Bhatt, Jonathan Babb, Steven Swanson, Michael Taylor. "The GreenDroid Mobile Application Processor: An Architecture for Silicon's Dark Future." *Micro, IEEE*. 31. no. 2 (2011): 8. http://0-ieeexplore.ieee.org.library.uark.edu/stamp/stamp.jsp?tp=&arnumber=5719585 (accessed November 28, 2011).
9. Borkar, Shekhar, and Andrew Chien. "The Future of Microprocessors." *Communications of the ACM*, May 2011. http://delivery.acm.org/10.1145/1950000/1941507/p67-borkar.pdf?ip=130.184.253.19&acc=OPEN&CFID=70693110&CFTOKEN=24930614&__acm__=1322543360_0ce33b6c13b6f2a6217558a820073ca9 (accessed November 28, 2011).
10. Wikipedia, "Moore's Law." Last modified November 29, 2011. Accessed November 29, 2011. http://en.wikipedia.org/wiki/Moore's_law